15-853:Algorithms in the Real World

Linear and Integer Programming I

- Introduction
- Geometric Interpretation
- Simplex Method
- Duality

Linear and Integer Programming

Linear or Integer programming

minimize $z = c^T x$ **cost** or **objective** function

subject to $Ax = b$ **equalities**

$x \geq 0$ **inequalities**

$c \in \Re^n$, $b \in \Re^m$, $A \in \Re^{n \times m}$

Linear programming:

$x \in \Re^n$ (polynomial time)

Integer programming:

$x \in Z^n$ (NP-complete)

Very general framework, especially IP

Related Optimization Problems

Unconstrained optimization NP-hard

$\min\{f(x) : x \in \Re^n\}$

Constrained optimization NP-hard

$\min\{f(x) : c_j(x) \leq 0 \text{ for } j \in J, c_i(x) = 0, i \in E\}$

Quadratic programming NP-hard

$\min\{ \frac{1}{2} x^T Q x + c^T x : a_j^T x \leq b_j, j \in J, a_i^T x = b_i, i \in E\}$

Convex programming

$\min\{f(x) : f_j(x) \leq b_j, j \in J, f, f_j \text{ are convex}\}$

Mixed Integer Programming NP-hard

$\min\{c^T x : Ax = b, x \geq 0, x_j \in Z, j \in J, x_r \in \Re, r \in R\}$

How important is optimization?

- 50+ special-purpose packages available
  - matlab, maple, mathematica, all have built-in solvers
  - MS Excel has a built-in LP solver.
  - Google's solver: Glop.
- 100+ books in the library, thousands of papers.
- 100s of companies
  - Two main ones: CPLEX, Gurobi
- Major airlines, delivery/trucking companies, manufacturers, internet companies… all make serious use of optimization.

Linear+Integer Programming Outline

Linear Programming

- General formulation and geometric interpretation
- Simplex method
- Ellipsoid method
- Interior point methods

Integer Programming

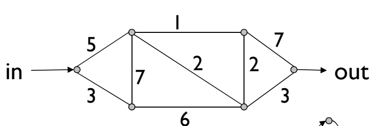
- Various reductions of NP hard problems
- Linear programming approximations
- Branch-and-bound + cutting-plane techniques
- Case study from Delta Airlines

Applications of Linear Programming

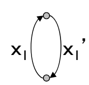
1. A substep in most integer and mixed-integer linear programming (MIP) methods

2. Used to approximate various NP-Hard problems
3. Selecting a mix: oil mixtures, portfolio selection
4. Distribution: how much of a commodity should be distributed to different locations.
5. Allocation: how much of a resource should be allocated to different tasks
6. Network Flows

Linear Programming for Max-Flow

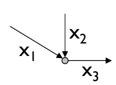


Create two variables per edge:



Create one equality per vertex:

$x_1 + x_2 + x_3' = x_1' + x_2' + x_3$



and two inequalities per edge:

$x_1 \le 3, \; x_1' \le 3$

add edge x_0 from **out** to **in**

maximize x_0

In Practice

In the "real world" most problems involve at least some integral constraints.

- Many resources are integral
- Can be used to model yes/no decisions (0-1 variables)

Therefore

"1. A subset in integer or MIP programming" is the most common application of LPs in practice

Algorithms for Linear Programming

- **Simplex** (Dantzig 1947)
- **Ellipsoid** (Khachian 1979)
  first algorithm known to be **polynomial time**
- **Interior Point**
  first practical polynomial-time algorithms
  - **Projective method** (Karmakar 1984)
  - **Affine Method** (Dikin 1967)
  - **Log-Barrier Methods** (Frisch 1977, Fiacco 1968, Gill et.al. 1986)

Many of the interior point methods can be applied to nonlinear programs – these not known to be poly. time

State of the art

1 million variables

10 million nonzeros

No clear winner between Simplex and Interior Point

- Depends on the problem
- Interior point methods subsuming more & more cases
- All major packages supply both
- which one can use parallelism better?

The truth: the sparse matrix routines make or break both methods.

The best packages are highly sophisticated.

State of the art

Which technique to use for solving LPs?
Pivot-based (simplex) vs interior-point.

Commercial solvers are very smart
often try to figure out which technique to use.

Many online resources and benchmarks: e.g.,
Decision Tree for Optimization Software

Formulations

There are many ways to formulate linear programs:

- **objective (or cost) function**
  maximize $c^T x$, or
  minimize $c^T x$, or
  find any feasible solution
- **(in)equalities**
  $Ax \leq b$, or
  $Ax \geq b$, or
  $Ax = b$, or any combination
- **nonnegative variables**
  $x \geq 0$, or not

Fortunately it is pretty easy to convert among forms

Formulations

The two **most common** formulations:

1

minimize $c^T x$
subject to $Ax \geq b$
$x \geq 0$

slack variables →

2

minimize $c^T x$
subject to $Ax = b$
$x \geq 0$

e.g.

$7x_1 + 5x_2 \geq 7$
$x_1, x_2 \geq 0$

y_1 →

$7x_1 + 5x_2 - y_1 = 7$
$x_1, x_2, y_1 \geq 0$

More on slack variables later.

Geometric View

A **polytope** in n-dimensional space

Each inequality corresponds to a half-space.

The "feasible set" is the intersection of the half-spaces.

This corresponds to a polytope

The optimal solution is at a corner.

Simplex moves around on the surface of the polytope

Interior-Point methods move within the polytope

Geometric View

minimize:

$z = -2x_1 - 3x_2$

subject to:

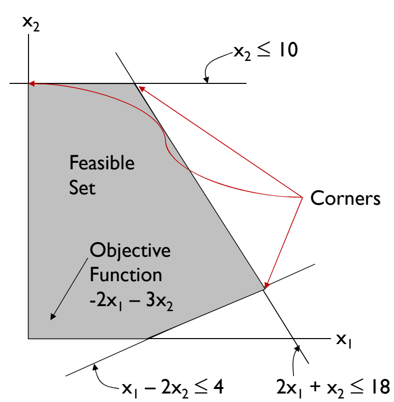
$x_1 - 2x_2 \leq 4$

$2x_1 + x_2 \leq 18$

$x_2 \leq 10$

$x_1, x_2 \geq 0$

An intersection of 5 halfspaces



Notes about higher dimensions

For n dimensions and no degeneracy

Each corner (extreme point) consists of:

– n intersecting (n-1)-dimensional **hyperplanes**
e.g. three intersecting 2d planes in 3d

– n intersecting **edges**

Each edge corresponds to moving off of one hyperplane (still constrained by n-1 of them)

 # Corners can be exponential in n (e.g. a hypercube)

Simplex will move from corner to corner along the edges

Simplex

The Simplex Method

Developed by George Dantzig in 1947.

Tjalling C. Koopmans, George Dantzig, and Leonid V. Kantorovich.
Koopmans and Kantorovich won the Nobel in 1975 for "optimal allocation of resources" using LPs.

The Simplex Method

Example on the board.

Simplex Concluding remarks

For dense matrices, takes O(n(n+m)) time per iteration
In practice, sparse methods are used for the iterations.

Can take an **exponential** number of iterations.

Some freedom in choice of entering plane
(called pivoting rule)
Sadly, most pivoting rules have bad examples with exponential number of iterations.

Still, in practice, it's a great algorithm!

Linear Programming Duality

15-853 Page22

Duality

Primal (P):

maximize $z = c^T x$

subject to $Ax \le b$

$x \ge 0$ (n equations, m variables)

Dual (D):

minimize $z = y^T b$

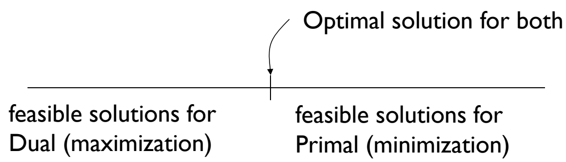
subject to $A^T y \ge c$

$y \ge 0$ (m equations, n variables)

Duality Theorem: if x is feasible for **P** and y is feasible for **D,** then $cx \le yb$
and at optimality $cx = yb$.

15-853 Page23

Duality (cont.)



Quite similar to duality of Maximum Flow and Minimum Cut.

Useful in many situations.

15-853 Page24

Duality Example

Primal:

maximize:

$z = 2x_1 + 3x_2$

subject to:

$x_1 - 2x_2 \le 4$

$2x_1 + x_2 \le 18$

$x_2 \le 10$

$x_1, x_2 \ge 0$

Dual:

minimize:

$z = 4y_1 + 18y_2 + 10y_3$

subject to:

$y_1 + 2y_2 \ge 2$

$-2y_1 + Y_2 + Y_3 \ge 3$

$y_1, y_2, y_3 \ge 0$

Solution to both is 38 ($x_1 = 4$, $x_2 = 10$), ($y_1 = 0$, $y_2 = 1$, $y_3 = 2$)

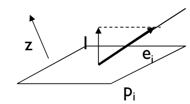
15-853 Page25

The Tableau View of Simplex (optional)

Optimality and Reduced Cost

The **Optimal** solution must include a corner.

The **Reduced cost** for a hyperplane at a corner is the cost of moving one unit away from the plane along its corresponding edge.

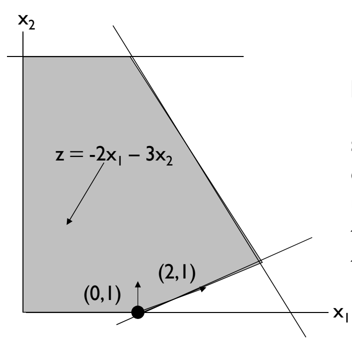


$r_i = z \cdot e_i$

For **minimization**, if all reduced cost are non-negative, then we are at an optimal solution.

Finding the most negative reduced cost is a heuristic for choosing an edge to leave on

Reduced cost example

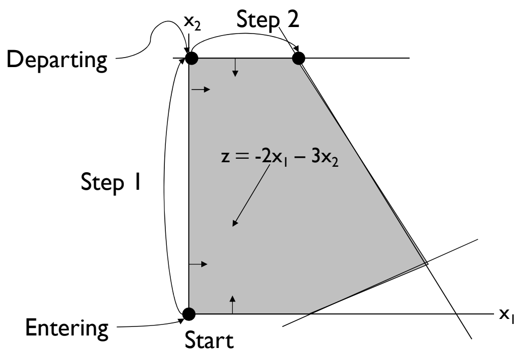


In the example the reduced cost of leaving the plane x_1 is $(-2,-3) \cdot (2,1) = -7$ since moving one unit off of x_1 will move us (2,1) units along the edge. We take the dot product of this and the cost function.

Simplex Algorithm

1. Find a **corner of the feasible region**
2. **Repeat**
   - A. For each of the n hyperplanes intersecting at the corner, calculate its **reduced cost**
   - B. If they are all non-negative, then **done**
   - C. Else, pick the most negative reduced cost
     This is called the **entering** plane
   - D. Move along corresponding edge (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane)
     The new plane is called the **departing** plane

Example



Simplifying

Problem:

- The $Ax \leq b$ constraints not symmetric with the $x \geq 0$ constraints.
  We would like more symmetry.

Idea:

- Make all inequalities of the form $x \geq 0$.

Use "slack variables" to do this.

Convert into form:

minimize	$c^T x$	
subject to	$Ax = b$	
	$x \geq 0$	

Standard and Slack Form

Standard Form

minimize	$c^T x$
subject to	$Ax \leq b$
	$x \geq 0$

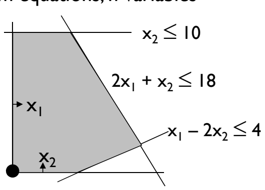
slack variables →

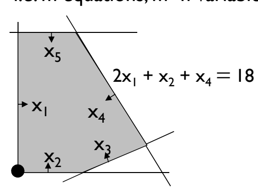
Slack Form

minimize	$c^T x'$
subject to	$A' x' = b$
	$x' \geq 0$

$|A| = m \times n$
i.e. m equations, n variables

$|A'| = m \times (m+n)$
i.e. m equations, m+n variables





Example, again

minimize:

$z = -2x_1 - 3x_2$

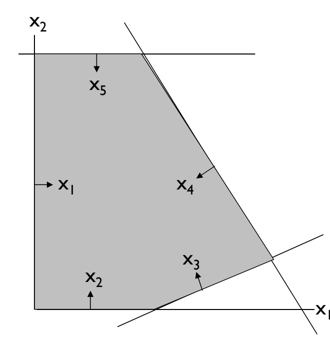
subject to:

$x_1 - 2x_2 + \underline{x_3} = 4$

$2x_1 + x_2 + \underline{x_4} = 18$

$x_2 + \underline{x_5} = 10$

$x_1, x_2, \underline{x_3}, \underline{x_4}, \underline{x_5} \geq 0$

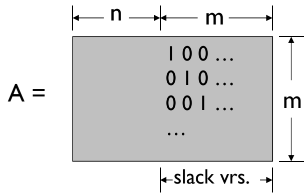


The equality constraints impose a 2d plane embedded in 5d space, looking at the plane gives the figure above

Using Matrices

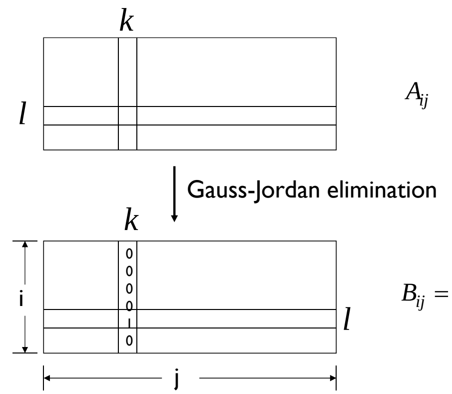
If before adding the slack variables A has size m x n then after it has size m x (n + m)

m can be larger or smaller than n



Assuming rows are independent, the solution space of Ax = b is a n dimensional subspace.

Gauss-Jordan Elimination



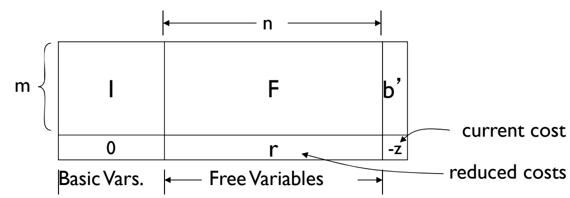
A_{ij}

$$B_{ij} = \begin{cases} A_{ij} - A_{lj}\dfrac{A_{ik}}{A_{lk}} & i \neq l \\ \dfrac{A_{lj}}{A_{lk}} & i = l \end{cases}$$

Simplex Algorithm, again

1. Find a **corner of the feasible region**
2. **Repeat**
   - A. For each of the n hyperplanes intersecting at the corner, calculate its **reduced cost**
   - B. If they are all non-negative, then **done**
   - C. Else, pick the most negative reduced cost
     This is called the **entering** plane
   - D. Move along corresponding line (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane)
     The new plane is called the **departing** plane

Simplex Algorithm (Tableau Method)



This form is called a **Basic Solution**

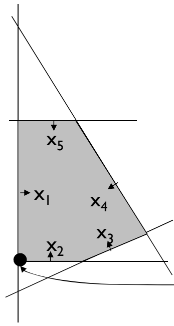
- the n "**free**" variables are set to 0
- the m "**basic**" variables are set to b'

A valid solution to Ax = b if reached using Gaussian Elimination

Represents n intersecting hyperplanes

If feasible (i.e. b' ≥ 0), then the solution is called a **Basic Feasible Solution** and is a corner of the feasible set

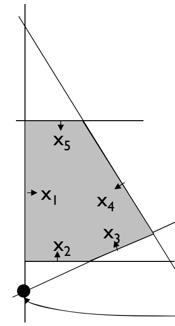
Corner



basic variables | free variables

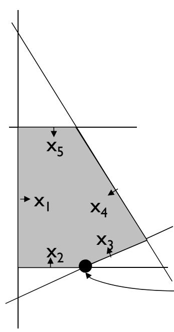
1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0
x_3	x_4	x_5	x_1	x_2	

Corner



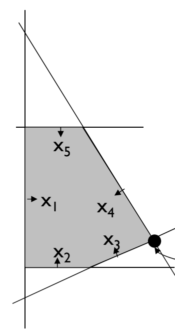
1	0	0	-.5	-1	-2
0	1	0	2.5	1	20
0	0	1	.5	1	12
0	0	0	-3.5	-3	-6
x_2	x_4	x_5	x_1	x_3	

Corner



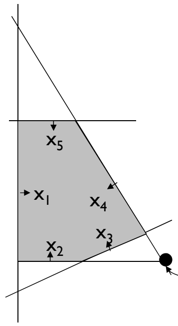
1	0	0	1	-2	4
0	1	0	-2	5	10
0	0	1	0	1	10
0	0	0	2	-7	8
x_1	x_4	x_5	x_3	x_2	

Corner



1	0	0	.2	.4	8
0	1	0	-.4	.2	2
0	0	1	.4	-.2	8
0	0	0	-.8	1.4	22
x_1	x_2	x_5	x_3	x_4	

Corner



1	0	0	.5	-2.5	-5
0	1	0	.5	.5	9
0	0	1	0	1	10
0	0	0	1	-2	18
x_3	x_1	x_5	x_4	x_2	

Note that in general there are n+m choose m corners

Simplex Method Again

Once you have found a basic feasible solution (a corner), we can move from corner to corner by swapping columns and eliminating.

ALGORITHM

1. Find a **basic feasible solution**
2. **Repeat**
   A. If r (reduced cost) ≥ 0 , DONE
   B. Else, pick column with most negative r
   C. Pick row with least positive b' /(selected column)
   D. Swap columns
   E. Use Gauss-Jordan elimination to restore form

Tableau Method

A. If r are all non-negative then **done**

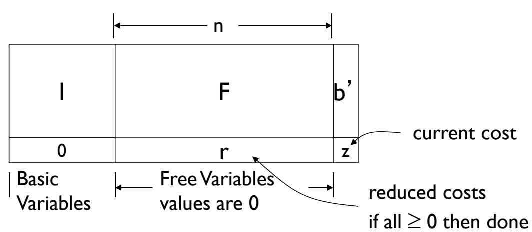


Tableau Method

B. Else, pick the most negative reduced cost
This is called the **entering** plane or variable

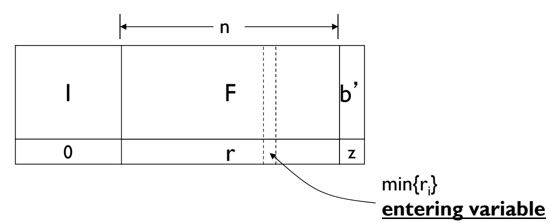
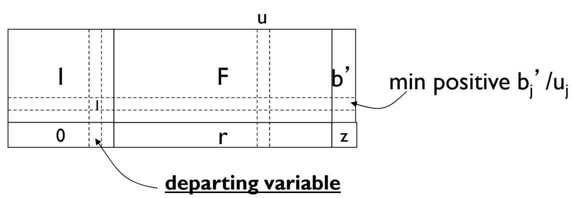


Tableau Method

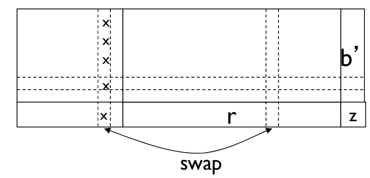
C. Move along corresponding line (i.e. leave that hyperplane) until we reach the next corner (i.e. reach another hyperplane)
The new plane is called the **departing** plane



15-853 Page46

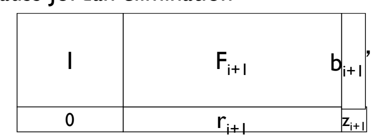
Tableau Method

D. Swap columns



No longer in proper form

E. Gauss-Jordan elimination



Back to proper form

15-853 Page47

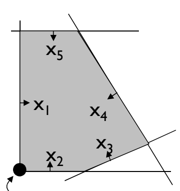
Example

x_1	x_2	x_3	x_4	x_5	
1	-2	1	0	0	4
2	1	0	1	0	18
0	1	0	0	1	10
-2	-3	0	0	0	0

$x_1 - 2x_2 + x_3 = 4$
$2x_1 + x_2 + x_4 = 18$
$x_2 + x_5 = 10$
$z = -2x_1 - 3x_2$

Find corner

1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0
x_3	x_4	x_5	x_1	x_2	



$x_1 = x_2 = 0$ (start)

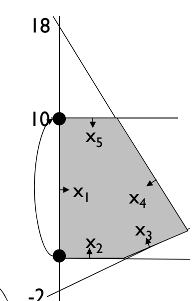
15-853 Page48

Example

1	0	0	1	-2	4
0	1	0	2	1	18
0	0	1	0	1	10
0	0	0	-2	-3	0
x_3	x_4	x_5	x_1	x_2	

						b_j/v_j
1	0	0	1	-2	4	-2
0	1	0	2	1	18	18
0	0	1	0	1	10	10
0	0	0	-2	-3	0	
x_3	x_4	x_5	x_1	x_2		

min positive



15-853 Page49

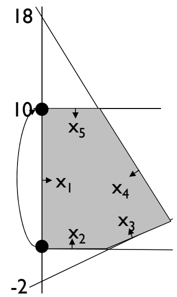
Example

swap

1	0	-2	1	0	4
0	1	1	2	0	18
0	0	1	0	1	10
0	0	-3	-2	0	0
x_3	x_4	x_2	x_1	x_5	

1	0	0	1	2	24
0	1	0	2	-1	8
0	0	1	0	1	10
0	0	0	-2	3	30
x_3	x_4	x_2	x_1	x_5	

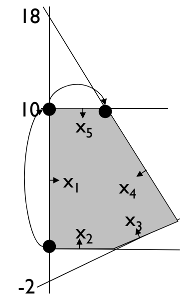
Gauss-Jordan Elimination



Example

1	0	0	1	2	24
0	1	0	2	-1	8
0	0	1	0	1	10
0	0	0	-2	3	30
x_3	x_4	x_2	x_1	x_5	

1	0	0	1	2	24	24
0	1	0	2	-1	8	4
0	0	1	0	1	10	1
0	0	0	-2	3	30	
x_3	x_4	x_2	x_1	x_5		



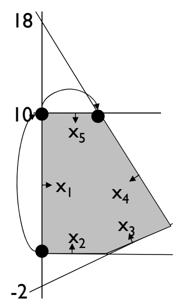
Example

swap

1	1	0	0	2	24
0	2	0	1	-1	8
0	0	1	0	1	10
0	-2	0	0	3	30
x_3	x_1	x_2	x_4	x_5	

1	0	0	-.5	2.5	20
0	1	0	.5	-.5	4
0	0	1	0	1	10
0	0	0	1	2	38
x_3	x_1	x_2	x_4	x_5	

Gauss-Jordan Elimination



Simplex Concluding remarks

For dense matrices, takes O(n(n+m)) time per iteration
In practice, sparse methods are used for the iterations.

Can take an **exponential** number of iterations.

Some freedom in choice of entering plane
(called pivoting rule)
Sadly, most pivoting rules have bad examples with exponential number of iterations.

Still, in practice, it's a great algorithm!

Duality

Primal (P):

maximize $z = c^T x$

subject to $Ax \le b$

$x \ge 0$ (n equations, m variables)

Dual (D):

minimize $z = y^T b$

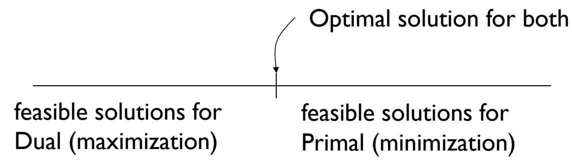
subject to $A^T y \ge c$

$y \ge 0$ (m equations, n variables)

Duality Theorem: if x is feasible for **P** and y is feasible for **D,** then $cx \le yb$ and at optimality $cx = yb$.

15-853 Page54

Duality (cont.)



Quite similar to duality of Maximum Flow and Minimum Cut.

Useful in many situations.

15-853 Page55

Duality Example

Primal:

maximize:

$z = 2x_1 + 3x_2$

subject to:

$x_1 - 2x_2 \le 4$

$2x_1 + x_2 \le 18$

$x_2 \le 10$

$x_1, x_2 \ge 0$

Dual:

minimize:

$z = 4y_1 + 18y_2 + 10y_3$

subject to:

$y_1 + 2y_2 \ge 2$

$-2y_1 + Y_2 + Y_3 \ge 3$

$y_1, y_2, y_3 \ge 0$

Solution to both is 38 ($x_1 = 4$, $x_2 = 10$), ($y_1 = 0$, $y_2 = 1$, $y_3 = 2$)

15-853 Page56