Consider QS(M) (distinct keys)
1) pick random a ∈ M
2) split M : $S < a < L$ (|M| - 1 comparisons)
3) return QS(S) * a * QS(L)

Goal: Expect # comparisons

Consider dart game:
Init: empty board

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

While non-empty square
pick random empty sq
Cost = # empty sqs to left & right of dart.

Claim: Expect cost of dart game = Expect cost QS.
Backwards game:

Init: full board

While 3 dart remove random dart.
Cost: # empty Ds left & right.

Claim: \[ \text{Expect cost DG} = \text{Expect cost BW DG} \]

Analysis backwards game

Assume i darts on board

\[ T_i = \text{Expected cost to remove random dart} \]

Total Cost = \[ \sum \text{cost d}_i \]

\[ \leq 2(n-i) \]

\[ \frac{T_i}{i} \leq \frac{2n}{i} \leq \frac{2n}{i} \]

\[ E(DG) = \sum T_i \leq \sum \frac{an}{i} = 2n H_n \]

\[ = O(n H_n) \]
Point Location Prob

Trapezoidal Decomp

Def Planar Straight Line Graph (PSLG) \( G = (V, E) \)

Today's Design assumption: no 2 vertices with same \( x \).

N-line seg 2N+3 vertices

Point location Prob

Input: \( G = (V, E) \) PSLG

Preprocess o

Query o Given \( P \) find face containing \( P \).
Simple 2D Alg

1) Sort $V$ by $x$-value
2a) Decompose $G$ in slabs
   b) sort each slab
3) Query $P_0$
   a) find slab containing $P_0$, $O(\sqrt{n})$
   b) find $P$ in slab, $O(\log n)$

$n/2$ slabs of size $n/2$!

$\mathcal{O}(n^2)$ answers!
Trap Decomp

Input: n-line seg (no two endpoints with same x)

Output:

a) Add bounding box
b) extend up & down each endpoint until it hits a seg.
Lemma

a) at most \(6n+1\) vertices

b) at most \(3n+1\) traps

a) 4 on bounding box

2n input

2(2n) intersections

b) each trap has unique left endpoint

each right endpoint "sees" one trap

each left "\(n\)", \(n\) \(2n\) traps

"sees" one trap

3n+1
Point location on Trip decoy

Assume order $s_1, s_2$
Random Incremental

Alg Trap Map (S)

Input: Segments S = S₁, ..., Sₙ

Output:
1) Trap Map T
2) Point Location DS D

1) Init: Build bounding box B & D for B
2) Randomly order S₁, ..., Sₙ
3) for i = 1 to n
   a) Find trap containing left-endpoint of Sᵢ
   b) Stitch in Sᵢ from left-to-right
   c) Update D

Following pointers:
1) left-vertical (Δ)
2) right-vertical (Δ)
3) left-trap (edge)
4) right-trap (edge)
Thm Trap Map

1) $O(n \log n)$ expected time to build
2) $|D| = O(n)$ expected
3) Search time is $O(\log n)$ expected.

\underline{Note} 1: Expectation is over $n!$ orderings of $S$.
\underline{Note} 2: Trap of $S_i \subseteq S$, $\mathbb{E}(S_i)$ is indep of order of $S_i$.
\underline{Note} 3: Each update of $90$ increases depth $\leq 3$.

Fix $g \in \mathbb{B}, bb$.

Consider random variable $X$:

$$X = \text{depth}(g, D)$$

Goal: Bound $E(X)$

Consider $X_i = \text{# levels added when adding } i\text{th edge}$.

Note $X = \sum X_i$
Thus \( E(X) = \sum E(X_i) \).

Note \( X_i \leq 3 \)

Consider fixed subset \( S_i \subseteq S \mid |S_i| = i \).

Let \( \Delta g(S_i) = \text{Trap containing } g \in T(S_i) \).

\[ P_i = \text{Prob} \left[ \Delta g(S_i) \neq \Delta g(S_{i-1}) \right] \]

Thus \( E(X_i) \leq 3P_i \)

Claim \( P_i \leq 4/i \)

Use Backward Analysis!

Of \( i \) seg remove one at random

Consider \( \Delta g(S_i) \)

\[ S_1 \quad \Delta g(S_i) \]

\[ S_2 \quad S_3 \quad S_4 \]

\[ S_2 \quad S_3 \quad \quad S_4 \]

Only removing \( S_1 \) to \( S_4 \) change \( \Delta g(S_i) \)
\[ E(X_i) \leq \frac{3 \cdot 4}{i} \quad \text{note: midp of set } S_i \]

\[ E(X) \leq 12 H_n \quad \text{the nth harmonic number!} \]
2) The size of $O$

$\text{\# nodes} = \text{\# Traps} + \text{\# internal nodes}$

$\text{\# Traps} = O(n)$ (Thm)

$k_i = \text{\# new traps at time } i$

$k_{i-1} = \text{\# internal nodes}

\underline{Backwards analysis}$

$k_i = \text{\# traps removed when random seed}$

$\text{\# Traps} = 3i+1$

$p_i \leq \frac{4}{i}$

$E(k_i) \leq (3i+1) \left(\frac{4}{i}\right) = O(1)$

$E(1D1) = \sum E(k_i) = O(n)$
1) Build Time

a) Expected time to find left endpoint
   \[O(\log i)\]

b) Expected work to stitch in seq in
   \[O(1)\]

Total expected work \[O(\sum_{i=1}^{n} i) = O(n \log n)\]