Recitation 2

Solving Recurrences

2.1 Announcements

- *ParenLab* is due **Friday at 5:00pm**. It is worth 150 points.
- *SkylineLab* will be released Friday and will be due the following Friday.

2.2 The Tree Method

The cost analysis of our algorihms usually comes down to finding a closed form for a recurrence. Using the tree method to derive the closed form consists of finding a cost bound for each level of the recursion tree and then summing the costs over the levels.

Task 2.1. *Using the tree method, solve the following recurrences:*

$$f(n) = 4f\left(\frac{n}{2}\right) + n^2$$
$$f(n) = 2f(n-1) + 1$$
$$f(n) = 2f\left(\frac{n}{2}\right) + n^{\frac{1}{4}}$$

2.3 The Brick Method

If the cost at every successive level is a multiplicative factor away from the cost of the previous level, we can use the brick method. First, determine whether the recurrence conforms to one of the three cases below and then apply the next step for that case. Otherwise, use the tree or the substitution method.

Definition 2.2. ***Balanced*** *The cost of every level is roughly equal. With a maximum cost of* L *and* d *levels, the total cost will be* $O(dL)$.

Root Dominated *Each level is a constant factor smaller than the previous level. With a cost of* L *at the root, the total cost will be* $O(L)$.

Leaf Dominated *Each level is a constant factor larger than the previous level. With a cost of* L_d *at the bottom-most* (d*th) level, the total cost will be* $O(L_d)$.

Task 2.3. *Solve the following recurrences using the brick method:*

$$f(n) = 2f\left(\frac{n}{4}\right) + \sqrt{n}$$
$$f(n) = f(\sqrt{n}) + \log n$$
$$f(n) = 2f(\sqrt{n}) + 1$$

2.4 The Substitution Method.

We can also use mathematical induction to solve recurrences. If you want to go via this route (and you don't know the answer a priori), you'll need to guess the answer first and check it. Since this technique relies on guessing an answer, you can sometimes fool yourself by giving a false proof. The following are some tips:

1. Spell out the constants. Do not use big-O—we need to be precise about constants, so big-O makes it super easy to fool ourselves.
2. Be careful that the induction goes in the right direction.
3. Add additional lower-order terms, if necessary, to make the induction go through.

Task 2.4. *Solve the following recurrence using (strong) induction:*

$$W(n) = 2W\left(\frac{n}{2}\right) + O(n)$$

2.5 Additional Exercises

Exercise 2.5. *There is a well known deterministic linear-work algorithm for finding the kth smallest value of a set of values. It uses the median of medians as the pivot. (The median value is the value v such that if the values are sorted, v would be in the middle). You don't need to understand why the algorithm works, but to be able to analyze its costs based on a description of its steps:*

1. *If the input has 5 or fewer values, find the median by brute force, otherwise:*
2. *Group the input into $n/5$ groups of 5 and find the median of each group in parallel.*
3. *Find the median of the $n/5$ medians recursively. Call this p.*
4. *Use p to filter out $3/10^{th}$s of the values in $\Theta(n)$ work and $\Theta(\log n)$ span.*
5. *Recurse on the remaining $7/10^{th}$s of the values.*

Task 2.6.

Write down recurrences for work and span.

Solve the recurrence for work in terms of Θ.

Solve the recurrence for span in terms of Θ. Warning: this is pretty hard.

.