CS 5633 Analysis of Algorithms – Spring 14

Exam 2

NAME: Solution

- This exam is closed-book and closed-notes, and electronic devices such as calculators or computers are not allowed. You are allowed to use a cheat sheet (half a single-sided letter paper).
- Please try to write legibly – if I cannot read it you may not get credit.
- **Do not waste time** – if you cannot solve a question immediately, skip it and return to it later.

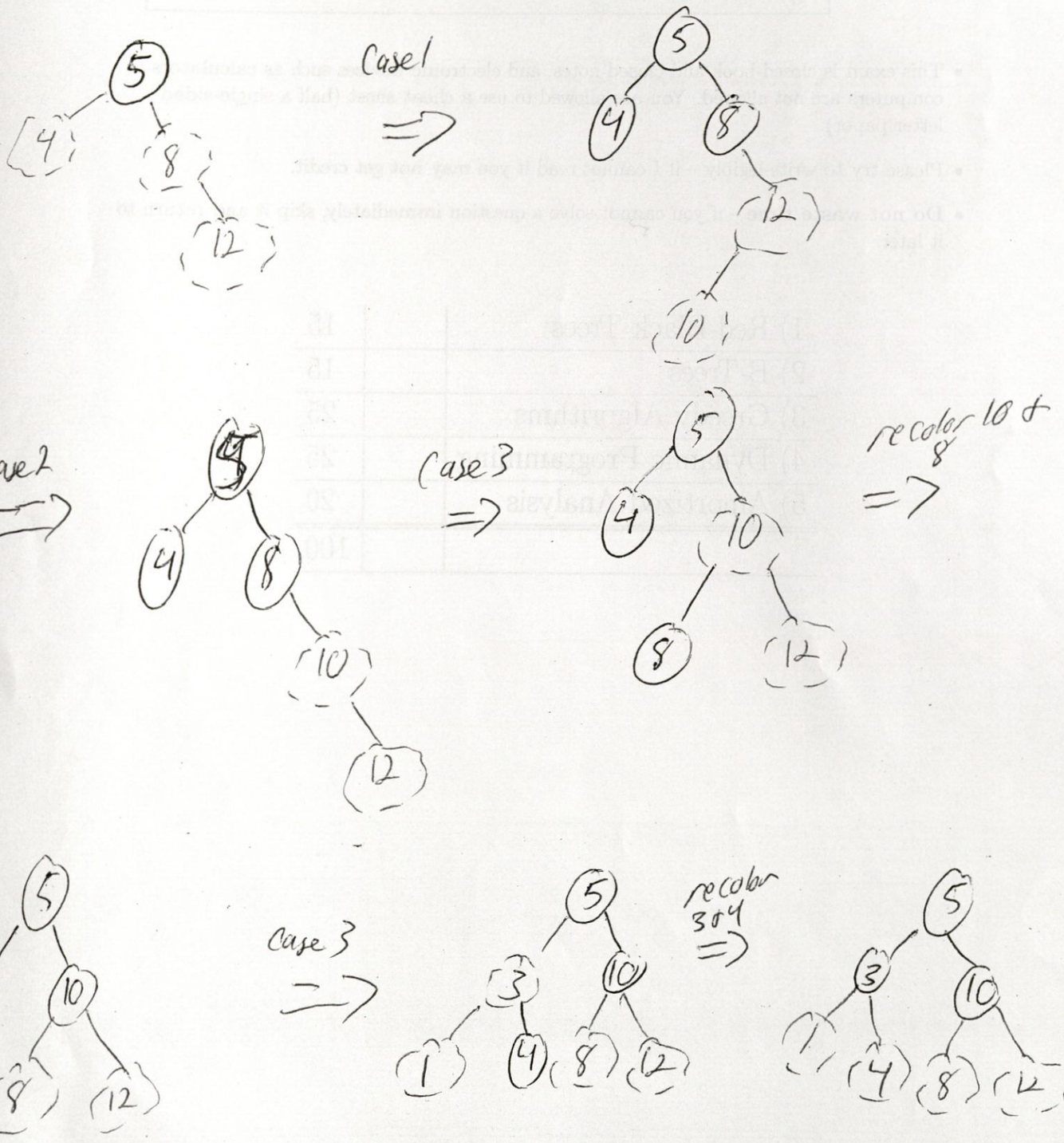
1) Red-Black Trees		15
2) B-Trees		15
3) Greedy Algorithms		25
4) Dynamic Programming		25
5) Amortized Analysis		20
		100

1 Red-Black Trees (15 Points)

Consider making the following sequence of inserts into an initially-empty red-black tree.

$$5, 8, 4, 12, 10, 3, 1$$

Show the tree before and after each rotation or recoloring (i.e. you do not need to draw it after each insertion if the insertion does not force a rotation or recoloring).

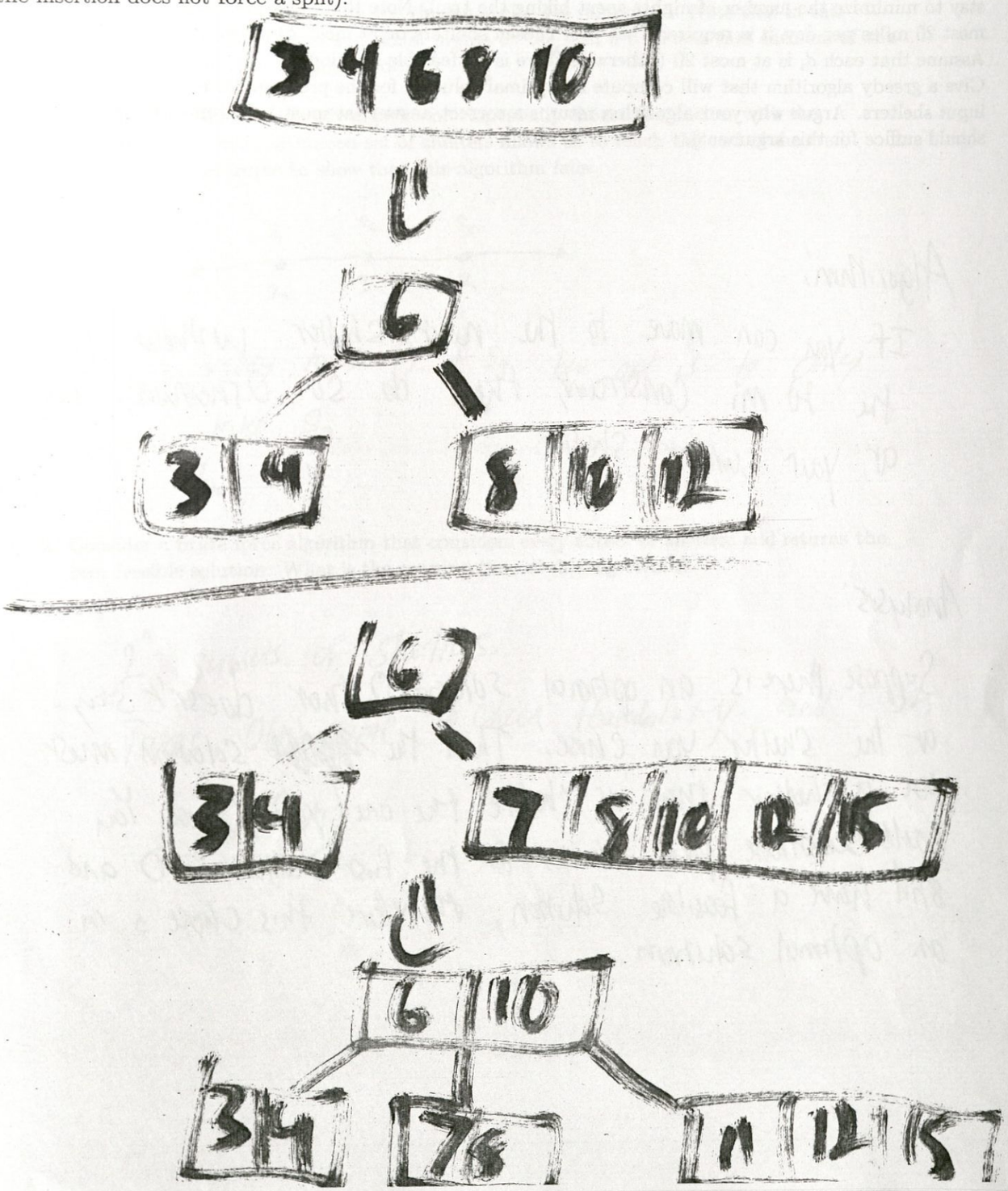


2 B-Trees (15 Points)

Consider making the following sequence of inserts into an initially-empty B-tree with $t = 3$.

$$3, 6, 10, 4, 8, 12, 15, 7, 11$$

Show the tree before and after each split (i.e. you do not need to draw it after each insertion if the insertion does not force a split).



3 Greedy Algorithms (25 Points)

Suppose you are going to go on a hike along the 2,000+ mile Appalachian Trail. Suppose you can hike at most 20 miles per day, and therefore will have to spend many nights camping along the trail. The trail has n shelters $s_1, s_2, \ldots, s_n$ along the way, and you are required to spend each night at a shelter. For each i from 1 to $n-1$, let d_i denote the distance between shelter s_i and shelter s_{i+1}. Your goal is to compute a sequence of shelters at which you will stay to minimize the number of nights spent hiking the trail. Note that since you can hike at most 20 miles per day, it is required that your chosen shelters be at most 20 miles apart. Assume that each d_i is at most 20 (otherwise there is no feasible solution).
Give a greedy algorithm that will compute an optimal solution for the problem for any set of input shelters. Argue why your algorithm returns a correct answer (at most 5 sentences or so should suffice for this argument).

Algorithm:

If you can move to the next shelter without violating the 20 mi constraint, then do so. Otherwise, stay at your current shelter.

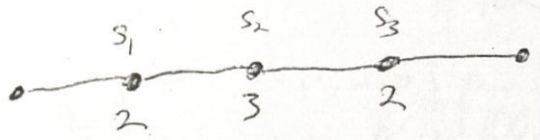
Analysis:

Suppose there is an optimal solution O that doesn't stay at the shelter you chose. Then the solution must use a shelter that is "before" the one you chose. You could substitute your shelter for the first shelter in O and still have a feasible solution, therefore this choice is in an optimal solution.

4 Dynamic Programming (25 Points)

This problem is similar to the greedy algorithms problem (you should read that question before reading this problem). We again want to hike the Appalachian Trail and can hike at most 20 miles per day. There again are n shelters $s_1, \ldots, s_n$, but now suppose there is a positive cost c_i that is charged to hikers for using shelter s_i. For simplicity, assume that "adjacent" shelters are 10 miles apart. The goal is to compute a sequence of shelters along the trail such that the sum of the costs of the chosen shelters is minimized. Note that in this situation, we are willing to spend additional nights on the trail if it means that the sum of the costs of our chosen shelters is minimized.

1. Consider the following greedy algorithm. Repeatedly choose the shelter with the minimum cost until our chosen set of shelters allows us to reach the end of the trail. Give a counterexample to show that this algorithm fails.



Greedy takes s_1 & s_3 but opt is to only take s_2.

2. Consider a brute force algorithm that considers every subset of shelters and returns the best feasible solution. What is the running time of this algorithm?

2^n subsets of shelters.

Takes $O(n)$ time to check feasability and cost.

$O(2^n \cdot n)$

3. Let $a[i]$ denote the optimal cost to hike to shelter s_i. Note that this does not necessarily mean that we must stay at shelter s_i. For example, if we stay at shelter s_{i-1} then we can walk to shelter s_i in an additional 10 miles and would not necessarily have to stay at shelter s_i.

   Suppose we have 5 shelters with the following costs: $A = \{5, 10, 4, 2, 8\}$. What is $a[1], a[2], a[3], a[4]$, and $a[5]$?

5 $a[1] = 0$ ⊗ × × × ×

10 $a[2] = 0$ ⊗ ⊗ × × ×

4 $a[3] = 5$ ✓ ⊗ ⊗ × × $\min\{0+5, 0+10\} = 5$

2 $a[4] = 9$ ✓ ⊗ ✓ ⊗ × $\min\{0+10, 5+4\} = 9$

8 $a[5] = 9$ ✓ ⊗ ✓ ⊗ ✗ $\min\{5+4, 9+2\} = 9$

4. Give a recursive definition for $a[i]$. Do not forget the base case.

$a[1] = 0$

$a[2] = 0$

$$a[i] = \min \begin{cases} c_{i-1} + a[i-1] \\ c_{i-2} + a[i-2] \end{cases} \quad \text{if } i \geq 3$$

5 Amortized Analysis (20 Points)

Recall the dynamic array data structure that was asked in class. If our data structure currently has size k, then we double the size to $2k$ once we receive operation $k+1$ and copy all of the elements from the old data structure to the new one. It is natural to view this doubling as quite wasteful when k is large. Suppose instead of always doubling the size of the array, we instead increase its size by 100 when needed (i.e. going from k to $k+100$ rather than $2k$). This question will consider the amortized cost of an operation in this data structure.

1. Perform an aggregate analysis to obtain a bound on the sum of the cost of n operations.

i	1	2	3	4	5	6	--	100	101	102 ...
size$_i$	1	101	101	101	101	101	--	101	101	201
actual	1	2	1	1	1	1	--	1	1	102

$$\text{Total} \le n + \sum_{j=0}^{n/100-1} (100j+1) = n + \left(100 \sum_{j=0}^{\frac{n}{100}-1} j\right) + \frac{n}{100} - 1$$

$$= n + \frac{n}{100} - 1 + \frac{\left(\frac{n}{100}-1\right)\left(\frac{n}{100}\right)}{2} < 3n^2$$

2. Use the accounting method to determine the amortized cost of a single operation.

We will show for $\hat{c}_i = 3n$. Need to prove for all "expensive" operations $k \le n$.

On k^{th} step: $\boxed{\sum_{i=1}^{k} \hat{c}_i = 3nk.}$

$$\sum_{i=1}^{k} c_i = n + 100 \sum_{j=0}^{\frac{k}{100}} j = n + 100 \frac{\left(\frac{k}{100}\right)\left(\frac{k}{100}+1\right)}{2} < n + 2k^2$$

Since $k \le n$, we have $\sum_{i=1}^{k} \hat{c}_i \ge \sum_{i=1}^{k} c_i$.