Sample Midterm Exam Solutions

1 Short answers [40 points]

(10) Part A. Solve the following recurrence by finding a $\Theta()$ bound for $T(n)$. Assume that $T(n)$ is 1 for $n \leq 1$, and otherwise is defined by the recurrence.

1. $T(n) = 15T(n/4) + 5n^2 + 7n$

Using the master theorem. $a = 15, b = 4, f(n) = 5n^2 + 7n = \Theta(n^2)$ so, $i = 2$

Case 3 applies since $a < b^i$, $T(n) = \Theta(n^2)$

---

(15) Part B. For the closest point algorithm we discussed in class, suppose we knew that every time we divide some number of points k, when we execute the combine step (where we use the "strip" points which are within d of the dividing line) at most $\sqrt{k}$ points are within d of the dividing line.

i) For this setting write a recurrence relation for $W(n)$, the maximum number of distance computations (where we compute the distance between two pairs of points). DO NOT use big O notation. Try and give the constants as precisely as you can. Note that you can ignore work other than distance computations.

In the regular version, each point on the left may be compared to 6 other points during the combine phase. In this problem, at most $\sqrt{n}/2$ points will be compared to six other points.

$W(n) = 2W(n/2) + 3\sqrt{n}$

You are told to give constants as precisely as you can, therefore is not completely correct to let $f(n) = O(\sqrt{n})$ Also those who put $f(n) = O(n)$ didn't receive much credit because this is the solution for the regular version from the book.

ii) Solve your recurrence to within Θ accuracy.

Using the master's, $W(n) = \theta(n)$

(15) Part C. Suppose we are using hashing with chaining to store n items in a table of size $m = 3n$. After hashing we observe that $n/4$ items are in each of two chains, while the other $n/2$ items had no collisions (so are each in their own chain).

i) If all items are equally likely to be looked up, what is the expected number of probes used in a successful search (where we use k probes if the key we find requires us to look at k items in the chain).

1/2 chance of one probe (Items with no collisions)

1/2 chance of searching through n/4 items.

Average search in the chain $= \frac{1+2+3+...+n/4}{n/4} = \frac{(n/4+1)(n/4)}{n/8} = (n/8 + .5)$

Expected probes $= 1/2 + (n/8 + .5)(1/2) = n/16 + 3/4$

There were many answers using α from the book. This is only good if it is uniform hashing, half the items hashing into two buckets is not uniformed hashing. Also, some wrote equations without any explanations. If these equations are wrong, I cannot tell if you were on the right track or not. Give explanations on what the equations mean so if they are wrong, you can still get partial credit.

ii) If all positions are equally likely to start an unsuccessful search, what is the expected number of probes for an unsuccessful search (where searching a chain with k items takes $k+1$ probes)?

We have 3n possible hash locations where our unsuccessful search will begin. 2 of these require $n/4+1$ probes, $n/2$ require 2 probes, and the remaining require one probe (Theses are the empty cells.)

$$\frac{2\cdot(n/4+1)+(n/2)\cdot 2+(3n-n/2-2)\cdot 1}{3n}$$

$$\frac{n/2+2+n+3n-n/2-2)}{3n}$$

$$\frac{4}{3}$$

2 Short answers Midterm 2 version [40 points]

(10) Part A. Solve the following recurrence by finding a $\Theta()$ bound for $T(n)$. Assume that $T(n)$ is 1 for $n \leq 1$, and otherwise is defined by the recurrence.

1. $T(n) = 15T(n/3) + 5n^2 + 7n$

Using the master theorem. $a = 15, b = 3, f(n) = 5n^2 + 7n, \quad i = 2$

Case 1 applies since $a > b^i$, $T(n) = \Theta(n^{log_b a}) = \Theta(n^{log_3 15})$

---

(15) Part B. For the closest point algorithm we discussed in class, suppose we knew that every time we execute the combine step (where we use the "strip" points which are within d of the dividing line) at most half the points are within d of the dividing line.

i) For this setting write a recurrence relation for $W(n)$, the maximum number of distance computations (where we compute the distance between two pairs of points). DO NOT use big O notation. Try and give the constants as precisely as you can. Note that you can ignore work other than distance computations.

In the regular version, each point may be compared to 6 other points during the combine phase. In this problem, at most half the $n/2$ points (so $n/4$) will be compared to six other points.

$W(n) = 2W(n/2) + 3n/2$

You are told to give constants as precisely as you can. Those who put $f(n) = n$ didn't receive much credit because this is the solution for the regular version from the book. The answer needed to show the importance of the $n/2$.

ii) Solve your recurrence to within Θ accuracy.

Using the master's, $W(n) = \theta(nlogn)$

(15) Part C. Suppose we are using hashing with chaining to store n items in a table of size $m = 2n$. After hashing we observe that $n/2$ items are in one chain, while the other $n/2$ items had no collisions (so are each in their own chain).

i) If all items are equally likely to be looked up, what is the expected number of probes used in a successful search (where we use k probes if the key we find requires us to look at k items in the chain).

1/2 chance of one probe (Items with no collisions)

1/2 chance of searching through n/2 items.

Average search in the chain $= \frac{1+2+3+...+n/2}{n/2} = \frac{(n/2+1)(n/2)}{n/2} = (n/2+1)$

Expected probes $= 1/2 + (n/2+1)(1/2) = n/4 + 1$

There were many answers using α from the book. This is only good if it is uniform hashing, half the items hashing into one bucket is not uniformed hashing. Also, some wrote equations without any explanations. If these equations are wrong, I cannot tell if you were on the right track or not. Give explanations on what the equations mean so if they are wrong, you can still get partial credit.

ii) If all positions are equally likely to start an unsuccessful search, what is the expected number of probes for an unsuccessful search (where searching a chain with k items takes $k+1$ probes)?

We have 2n possible hash locations where our unsuccessful search will begin. 1 of these require $n/2+1$ probes, $n/2$ require 2 probes, and the remaining require one probe (Theses are the empty cells.)

$$\frac{1\cdot(n/2+1)+(n/2)\cdot 2+(2n-n/2-1)\cdot 1}{2n}$$

$$\frac{n/2+1+n+2n-n/2-1)}{2n}$$

$$\frac{3}{2}$$

3 Coin Changing Varients [25 points]

Suppose we have an unlimited supply of pennies, dimes, and quarters but only two half-dollars. Given an amount N, we want to compute the minimum number of coins needed to make change of N cents in this setting.

Assume we have a program COINS such that COINS(N) returns the minimum number of coins to make change using pennies, dimes, quarters (but NOT using half-dollars) and takes $O(N)$ time.

Part A. Show that the greedy strategy of always using the largest coin is NOT optimal in this setting.

55: greedy: H, 5P = 6 coins
optimal: Q, 3D = 4 coins

105: greedy: 2H, 5P = 7 coins
optimal: H, Q, 3D = 5 coins

Part B. Describe an efficient algorithm to compute *Change(N)*, the smallest number of coins required to make change of N in this setting (that is with at most two half-dollars). N.

$Change(N) = min\{COINS(N), 1 + COINS(N - 50), 2 + COINS(N - 100)\}$

(this assumes that COINS (N) returns HUGE for $N < 0$)

Note that it would also be sensible to use half dollars until you run out or the remaining change is below 75 (since we proved greedy is optimal above 74 cents), then just use two calls to COINS if a half dollar is left, or 1 other wise).

Part C. Justify the correctness of your solution of part **B**.

The optimal solution has to either have 0, 1, or 2 half-dollars in it, the solution finds the best of the options and returns the minimum of those.

Part D. Analyze the worst case running time of your solution of part **B**.

Running time = 3θ(N) = θ(N)

4 Dynamic Programming [22 points]

We consider a varient of the knapsack problem. We have n regular items each with a value v_i and a weight w_i and a global weight limit W. You want to find a set of items whose total value is as large as possible, and the total cost is at most W. We also have three special items with weights $w_{n+1}, w_{n+2}, w_{n+3}$ and values $v_{n+1}, v_{n+2}, v_{n+3}$. We can use at most one of these three items in the solution (but we still have to obey the weight limit W for all items).

Part A. Describe how to find the best total value in this setting when all the items are fractional (if you use part of a special item, you cannot use part of another special item).

Be sure to justify your answer and give its running time.

We compute $r_i = v_i/w_i$ ratio for the n regular items and find the best set of items $1, 2, \ldots i$ with the highest r_j values using the regular greedy algorithm.

For each of the three special items, if its ratio is better than r_i, we replace item i by the special item, then $i-1, \ldots$ until we run out of the special item or we hit an item with ratio better than that of the special item.

We compare these three totals, and use the best one.

Each of the steps can be done in $\Theta(n)$ time, so total time is $\Theta(n)$.

To see correctness, note that we try each option (use special item 1, 2 and three), and use it in the best way (replacing worst ratio items with that item).

Note that we can't just use the special item with the best ratio. Since its weight may be smaller than another item (e.g. all regular items have $r_i = 1$, $W = 10$, and one special item has value 20 and weight 2 while another has value 60 and weight 10. The first item has higher ratio, the second is used in an optimal solution.

Part B. Describe how to find the best value in the 0-1 setting where you must take all or none of each item. As in **A** justify the correctness of your solution and give its running time.

Run the normal D.P. algorithm on the regular items. The final row $T[n, j]$ has the best value of each weight $j \leq W$ using all n items. Now for each special item, consider it as an $n+1$ item continuing the D.P. formulation. The maximum value in each of these three rows gives the best solution with the option of using that item. Thus the maximum of the three n_1 rows is the best solution.

Time is dominated by the initial D.P. $\Theta(nW)$.