CECS 528, Learning Outcome Assessment 7, Yellow, Fall 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit each solution on a separate sheet of paper.

Problems

LO3. Solve the following problems.

(a) Recall the steps of the `Minimum Distance Pair` algorithm. If $x = l$ is the vertical line that divides the problem instance into two equal halves, and point P of the data set lies to the left of the line $x = l - \delta$, then how do we know that the minimum distance pair cannot be (P, Q) where Q is a data point that lies to the right of $x = l$? Explain.

(b) For the Median-of-Five `Find-Statistic` algorithm, determine the value of the pivot M (at the top level of recursion) for

$$a = 75, 29, 16, 21, 52, 87, 4, 80, 78, 99, 11, 42, 14, 73, 15, 88, 36, 55, 91, 96, 18, 54, 9$$

and $k = 3$. Show work.

LO4. Answer the following.

(a) The FFT algorithm owes its existence to what two properties that are possessed by the nth roots of unity when n is even?

(b) Compute $\mathrm{DFT}_4^{-1}(6, -3, 1, -8)$ using the IFFT algorithm. Show the solution to each of the seven subproblem instances and, for each one, clearly represent it using DFT^{-1} notation and apply the formula for computing it. Show all work.

LO5. Answer the following with regards to a correctness-proof outline for Prim's algorithm.

(a) In the correctness proof of Prim's algorithm, suppose $T = e_1, \ldots, e_{n-1}$ are the edges selected by Prim's algorithm (in that order) and T_{opt} is an mst that uses edges $e_1, \ldots, e_{k-1}$, but for which $e_k \notin T_{\text{opt}}$. Then $T_{\text{opt}} + e_k$ has a cycle C. Explain why i) C enters T_{k-1}, Prim's tree after round $k - 1$, and why ii) C leaves T_{k-1}. In other words, explain why it is neither possible for C to lie completely inside of T_{k-1}, nor possible for C to lie completely outside of T_{k-1}.

(b) From part a, we know there must be at least one edge $e \in C$ for which i) $e \neq e_k$ and ii) e is incident with both a vertex in T_{k-1} and a vertex not in T_{k-1}. Explain why $w(e) \geq w(e_k)$.

(c) Based on part b, explain why $T_{\text{opt}} + e_k - e$ is a minimum spanning tree. In other words, why is $T_{\text{opt}} + e_k - e$ a tree, and why does it have a minimum cost?

LO6. For the weighted graph with edges

$$(a, e, 6), (b, e, 4), (c, e, 3), (c, d, 5), (d, f, 2), (e, f, 1),$$

Show how the disjoint-set data structure forest changes when processing each edge in Kruskal's sorted list of edges. When unioning two trees, use the convention that the root of the union is the root which has the *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are to be unioned, then the unioned tree should have root a.

LO7. Answer/Solve the following questions/problems.

(a) The dynamic-programming algorithm that solves the `Optimal Binary Search Tree` problem defines a recurrence for the function $\text{wac}(i, j)$. In words, what does $\text{wac}(i, j)$ equal? Hint: do *not* write the recurrence (see Part b).

(b) Provide the dynamic-programming recurrence for $\text{wac}(i, j)$.

(c) Apply the recurrence from Part b to the keys 1-5 whose respective weights are 70,90,50,20,30 Show the matrix of subproblem solutions and use it to provide an optimal binary search tree. For each subproblem solution, make sure to indicate the value of k that produced the solution.

Solutions

LO3. Solve the following problems.

(a) Recall the steps of the `Minimum Distance Pair` algorithm. If $x = l$ is the vertical line that divides the problem instance into two equal halves, and point P of the data set lies to the left of the line $x = l - \delta$, then how do we know that the minimum distance pair cannot be (P, Q) where Q is a data point that lies to the right of $x = l$? Explain.

Answer. If P lies to the left of line $x = l - \delta$, then P is more than δ away from the dividing line $x = l$ and so $d(P, Q) > \delta$ for all Q on or to the right of $x = l$. But from the two subproblem solutions, we know the minimum distance pair has a distance that is no greater than δ.

(b) For the Median-of-Five `Find-Statistic` algorithm, determine the value of the pivot M (at the top level of recursion) for

$$a = 75, 29, 16, 21, 52, 87, 4, 80, 78, 99, 11, 42, 14, 73, 15, 88, 36, 55, 91, 96, 18, 54, 9$$

and $k = 3$. Show work.

Solution. The medians of the five groups of 5 are 29, 80, 15, 88, and 18 with the median of these medians equal to 29. Therefore, $M = 29$.

LO4. Answer the following.

(a) The FFT algorithm owes its existence to what two properties that are possessed by the nth roots of unity when n is even?

Answer. The squares of the nth roots of unity give the $n/2$ roots of unity, and the nth roots of unity come in additive inverse pairs so that their squares produce only $n/2$ values (which happen to be the $n/2$ roots of unity).

(b) Compute $\mathrm{DFT}_4^{-1}(6, -3, 1, -8)$ using the IFFT algorithm. Show the solution to each of the seven subproblem instances and, for each one, clearly represent it using DFT^{-1} notation and apply the formula for computing it. Show all work.

Solution.

We have

$$\mathrm{DFT}_1^{-1}(6) = 6 \text{ and } \mathrm{DFT}_1^{-1}(1) = 1.$$

$$\mathrm{DFT}_2^{-1}(6, 1) = 1/2((6, 6) + (1, -1) \odot (1, 1)) = (7/2, 5/2).$$

$$\mathrm{DFT}_1^{-1}(-3) = -3 \text{ and } \mathrm{DFT}_1^{-1}(-8) = -8.$$

$$\mathrm{DFT}_2^{-1}(-3,-8) = 1/2((-3,-3) + (1,-1) \odot (-8,-8)) = (-11/2, 5/2).$$

$$\mathrm{DFT}_4^{-1}(4,-3,2,-1) = 1/2((7/2,5/2,7/2,5/2)+(1,-i,-1,i)\odot(-11/2,5/2,-11/2,5/2)) =$$

$$(-1, 5/4 - 5i/4, 9/2, 5/4 + 5i/4).$$

LO5. Answer the following with regards to a correctness-proof outline for Prim's algorithm.

(a) In the correctness proof of Prim's algorithm, suppose $T = e_1, \ldots, e_{n-1}$ are the edges selected by Prim's algorithm (in that order) and T_{opt} is an mst that uses edges $e_1, \ldots, e_{k-1}$, but for which $e_k \notin T_{\text{opt}}$. Then $T_{\text{opt}} + e_k$ has a cycle C. Explain why i) C enters T_{k-1}, Prim's tree after round $k-1$, and why ii) C leaves T_{k-1}. In other words, explain why it is neither possible for C to lie completely inside of T_{k-1}, nor possible for C to lie completely outside of T_{k-1}.

Answer. Since e_k is chosen in Round k it must be incident with one vertex of T_{k-1}. But it cannot include two vertices of T_{k-1} since then it would be an edge of T_{k-1}.

(b) From part a, we know there must be at least one edge $e \in C$ for which i) $e \neq e_k$ and ii) e is incident with both a vertex in T_{k-1} and a vertex not in T_{k-1}. Explain why $w(e) \geq w(e_k)$.

Answer. Since e is incident with one vertex of T_{k-1}, e was a candidate to be added to Prim's tree in Round k. But Prim chose e_k instead which means $w(e) \geq w(e_k)$.

(c) Based on part b, explain why $T_{\text{opt}} + e_k - e$ is a minimum spanning tree. In other words, why is $T_{\text{opt}} + e_k - e$ a tree, and why does it have a minimum cost?

Answer. Since e and e_k are both edges of the only cycle in $T_{\text{opt}}+e_k$, removing e eliminates the cycle while maintaining graph connectivity since traversing edge e can be replaced with traversing $C-e$. Finally, $T_{\text{opt}}+e_k-e$ still has minimum cost since $w(e_k) \leq w(e)$ (actually, their weights must be equal).

LO6. For the weighted graph with edges

$$(a,e,6), (b,e,4), (c,e,3), (c,d,5), (d,f,2), (e,f,1),$$

Show how the disjoint-set data structure forest changes when processing each edge in Kruskal's sorted list of edges. When unioning two trees, use the convention that the root of the union is the root which has the *lower* alphabetical order. For example, if two trees, one with root a, the other with root b, are to be unioned, then the unioned tree should have root a.

Solution.

LO7. Answer/Solve the following questions/problems.

(a) The dynamic-programming algorithm that solves the `Optimal Binary Search Tree` problem defines a recurrence for the function $\text{wac}(i, j)$. In words, what does $\text{wac}(i, j)$ equal? Hint: do *not* write the recurrence (see Part b).

Answer. $\text{wac}(i, j)$ the minimum weighted access cost of any tree that holds keys $i, \ldots, j$.

(b) Provide the dynamic-programming recurrence for $\text{wac}(i, j)$.

Answer.

$$\text{wac}(i,j) = \begin{cases} 0 & \text{if } j < i \\ w_i & \text{if } i = j \\ \min\limits_{i \le k \le j} (\text{wac}(i, k-1) + \text{wac}(k+1, j)) + \sum_{r=i}^{j} w_r & \text{otherwise.} \end{cases}$$

(c) Apply the recurrence from Part b to the keys 1-5 whose respective weights are 70,90,50,20,30 Show the matrix of subproblem solutions and use it to provide an optimal binary search tree. For each subproblem solution, make sure to indicate the value of k that produced the solution.

Solution. Entries of the form $\text{wac}(i, j)/k$ give the optimal weighted access cost , followed by the root k of the corresponding optimal tree.

i/j	1	2	3	4	5
1	70	230/2	330/2	390/2	500/2
2	0	90	190/2	250/2	350/3
3	0	0	50	90/3	170/3
4	0	0	0	20	70/5
5	0	0	0	0	30

The optimal tree root has key 2, while the optimal right subtree consists of the single branch 3,5,4.