CSE 3500 Algorithms and Complexity. Fall 2026

Home work 2 Solutions

1. We can see that the smallest element appears $2n-1$ times in C $(\min\{k_1,k_1\},\min\{k_1,k_2\},\ldots,\min\{k_1,k_n\},\min\{k_2,k_1\},\min\{k_3,k_1\},\ldots,\min\{k_n,k_1\})$. Similarly, the second smallest element apperas $2n-3$ times in C and the ith smallest element appears $2n-2i+1$ times in C. Now the median of C is the jth smallest element such that $\sum_{i=1}^{j} 2n-2i+1 = \frac{n^2}{2}$
   j can be obtained by solving the above equation, i.e., $j^2-2nj+\frac{n^2}{2}=0$.
   Now, the median of C can be obtained by finding the jth smallest element in S.
   Complexity $= O(n)$.

2. Sort the sets A and B using the radix sort algorithm. These sorts can be done in $O(n)$ time. Now compute $A \cap B$ in $O(n)$ time. (This can be done by merging A and B). If $A \cap B = \emptyset$ then the sets are disjoint. Total time taken is $O(n)+O(n)=O(n)$.

3. Let $(p_1,p_2)=(2,1),(w_1,w_2)=(x,1),m=1,x>1$. Then

$$\frac{F^*(I)}{\hat{F}(I)} = \frac{1}{2\frac{1}{x}} = \frac{x}{2} \to \infty \text{ when } x \to \infty$$

4. One possible minimum spanning tree has the following edges: $(1,3),(1,4),(2,3)$, and $(3,5)$. The total weight is 14. (You are supposed to give the detailed steps in either Prim's or Kruskal's algorithm).

5. At the beginning of the algorithm $dist[s]=0; dist[1]=2$; $dist[2]=15$; $dist[3]=dist[4]=dist[5]=\infty$.

   In stage 1, node 1 has the minimum $dist$ value and hence is inserted into the set S. Nodes 3 and 4 are neighbors of 1 and hence we have to check if the $dist$ values of thse nodes have to be modified. Since $dist[3] > dist[1]+W(1,3)$, we change $dist[3]$ to 8. Likewise we set $dist[4]=5$.

   In stage 2, node 4 has the minimum $dist$ value and hence is inserted into S. Nodes 2, 5, and 3 are neighbors of 4. The new $dist$ values of these nodes become: $dist[2]=10; dist[3]=6; dist[5]=10$.

   In stage 3, node 3 has the minimum $dist$ value and it becomes a part of S. All the neighbors of 3 are already in S and the $dist$ values of the nodes 2 and 5 do not change.

   In stage 4, we have two nodes both having the same dist value. We could pick one arbitrarily and insert it into S. Let 2 be this node. The $dist$ value of 5 does not change.

   In stage 5, the node 5 also enters S. Algorithm terminates then.

   Thus the shortest paths from s to the nodes 1, 2, 3, 4, and 5 are 2, 10, 6, 5, and 10, respectively.

6. As discussed in class, if $f_i(y)$ is the optimal profit for KNAP$(1,j,y)$, the recurrence relation for $f_i(y)$ is given by: $f_i(y)=\max\{f_{i-1}(y), f_{i-1}(y-w_i)+p_i\}$. Also, $f_0(y)=0$ for all non-negative values of y and $f_i(y)=-\infty$ when y is negative. From these relations we compute $f_0(y), f_1(y), f_2(y), f_3(y), f_4(y)$ for all $0 \le y \le 5$. These values are shown in the following table.

Function	$y = 0$	$y = 1$	$y = 2$	$y = 3$	$y = 4$	$y = 5$
f_0	0	0	0	0	0	0
f_1	0	10	10	10	10	10
f_2	0	10	15	25	25	25
f_3	0	10	15	25	35	40
f_4	0	10	15	25	35	40

For example, $f_3(5) = \max\{f_2(5), f_2(5-3)+25\} = \max\{25, 15+25\} = 40$. Also, $f_4(4) = \max\{f_3(4), f_3(4-2)+12\} = \max\{35, 15+12\} = 35$; and so on. Thus the optimal profit is 40.

7. Let A be the adjacency matrix of the graph (whose diagonal elements are zeros). It can be shown that $A^k(i,j) = 1$ iff there is a path from node i to node j of length exactly equal to k, for any $0 \leq k \leq (n-1)$. If there is a path at all from node i to node j in G, the shortest such path will be of length $\leq (n-1)$.
   Hence, $A^* = I + A + A^2 + \ldots + A^{n-1} = (I+A)^{n-1}$. Here, scalar addition corresponds to boolean or and scalar multiplication corresponds to boolean and.
   Now, $(I+A)^{n-1}$ can be computed by repeated squaring, i.e., $(I+A)^2, (I+A)^4, (I+A)^8$ etc.
   Complexity = Complexity of adding matrices I and A + Complexity of computing $(I + A)^{n-1} = O(M(n)\log n)$.