CS4335 Design and Analysis of Algorithms (Midterm, 2020)

Question 1. (20 points)

(a) (10 points) For the interval scheduling problem, the set of jobs (s_i, f_i) are as follows: (0, 2), (1, 3), (2, 6), (2, 4), (6, 9) (8, 12), (5, 8), and (6, 7).
Use a greedy algorithm to compute the maximum number of compatible jobs. You should give main steps. What is the running time of the greedy algorithm?

Answer:

(b) (8 points) For the interval partitioning problem, the set of lectures (s_i, f_i) are as follows: (0, 1), (0, 3), (1, 4), (2, 6), (2, 4), (4, 5), (3, 5) and (5, 8).
Use a greedy algorithm to compute the minimum number of classrooms to accommodate all the lectures. You should give main steps.

Answer:

(c) (2 points) For the interval partitioning problem given in (b), what is the depth of the problem?

Answer:

Question 2. (15 points)

(a) **(7 points)** Find the minimum spanning tree for the graph in Figure 1 using Kruskal's algorithm.
Answer:

(b) **(8 points)** Find the minimum spanning tree for the graph in Figure 1 using Prim's algorithm.

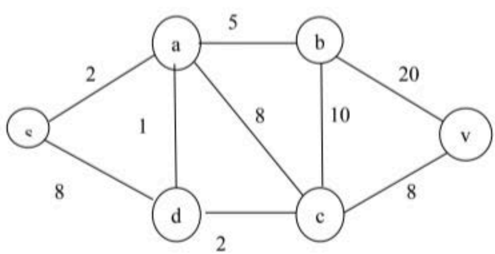


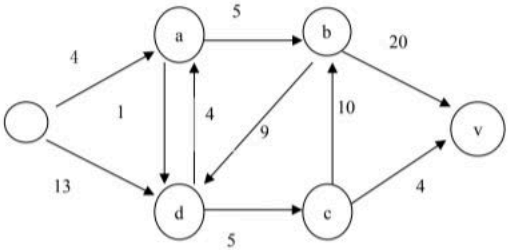
Figure 1.

Answer:

Question 3. (20 points)

(a) **(15 points)** Use Dijkstra's algorithm to compute a shortest path from *s* to *v* in the following graph. You should give main steps.

Answer:



(b) **(5 points)** Does Dijkstra's algorithm work for the case, where some edges can have negative weights? Why?

Answer:

Question 4 (15 points)

(a) (10 points) For the list: 2, 1, 5, 8, 9, 10, 4, 7, 6, 13, 14, and 11. Suppose we have sorted the two halves as list1: 1, 2, 5, 8, 9, 10; and list2: 4, 6, 7, 11, 13, 14. Calculate the number of inversions with one number in list1 and the other number in list2 using O(n) operations. Immediate steps are required.

(b) (1 points) Suppose $T(1)=1$, and $T(n)=T(n-1)+n$ for $k=2,3,4,$ ..... What is $T(n)$ in terms of big O notation?

(c) (1 points) Suppose $T(1)=1$, and $T(n)=T(n-1)+1$ for $k=2,3,4,$ ..... What is $T(n)$ in terms of big O notation?

(d) (1 points) Suppose $T(1)=1$, and $T(n)=T(n/2)+1$ for $k=2,3,4,$ ..... What is $T(n)$ in terms of big O notation?

(e) (1 points) Suppose $T(0)=1$, $T(1)=1$, and $T(n)=T(n-2)+\log_2 n$ for $k=2,3,4,$ ..... What is $T(n)$ in terms of big O notation?

(f) (1 points) Suppose $T(1)=1$, and $T(n)=T(n/3)+1$ for $k=2,3,4,$ ..... What is $T(n)$ in terms of big O notation?

Question 5. (15 points)

Given an array of $n \geq 2$ integers, say $[x(1),...,x(n)]$, we want to find the largest *difference* d, which is defined to be the max of $x(j)-x(i)$ over all $j > i$. For example, for $x = [22, 5, 8, 10, -3, 1]$, the largest *difference* $d = x(4)-x(2) = 10-5 = 5$.

(a)(**10 points**) Design a divide-and-conquer algorithm to solve the problem. The running time of your algorithm should be $O(n)$. (Only divide-conquer approach is acceptable. 0 mark for other methods.)

Question 1. (20 points)

(a) (10 points) For the interval scheduling problem, the set of jobs (s_i, f_i) are as follows: $(0, 2), (1, 3), (2, 6), (2, 4), (6, 9), (8, 12), (5, 8)$, and $(6, 7)$. Use a greedy algorithm to compute the maximum number of compatible jobs. You should give main steps. What is the running time of the greedy algorithm?

(b) (8 points) For the interval partitioning problem, the set of lectures (s_i, f_i) are as follows: $(0, 1), (0, 3), (1, 4), (2, 6), (2, 4), (4, 5), (3, 5)$ and $(5, 8)$. Use a greedy algorithm to compute the minimum number of classrooms to accommodate all the lectures. You should give main steps.

(c) (2 points) For the interval partitioning problem given in (b), what is the depth of the problem?

Solution 1

(a)
- Solution:

  Step 1. Sort all intervals in non-decreasing order by their finishing time f_i:
  $(0, 2), (1, 3), (2, 4), (2, 6), (6, 7), (5, 8), (6, 9), (8, 12)$.

  Step 2. Select the first job, and choose the rest jobs one by one if one is compatible with the former job:
  $(0, 2), (2, 4), (6, 7), (8, 12)$.

- Running time:
  Running time is $O(n \log n)$, where n is the number of intervals. Step 1 (sorting) requires the running time $O(n \log n)$. Step 2 (selecting) requires the running time $O(n)$ since each interval would be visited at most once. Hence, sorting is the dominated part which implies the running time is $O(n \log n)$.

(b) Step 1. Sort all intervals in non-decreasing order by their starting time s_i:
$(0, 1), (0, 3), (1, 4), (2, 4), (2, 6), (3, 5), (4, 5), (5, 8)$.

Step 2. Schedule the lectures (intervals) one by one, don't use the new classroom unless necessary. One possible interval partition can be found in Figure 1.

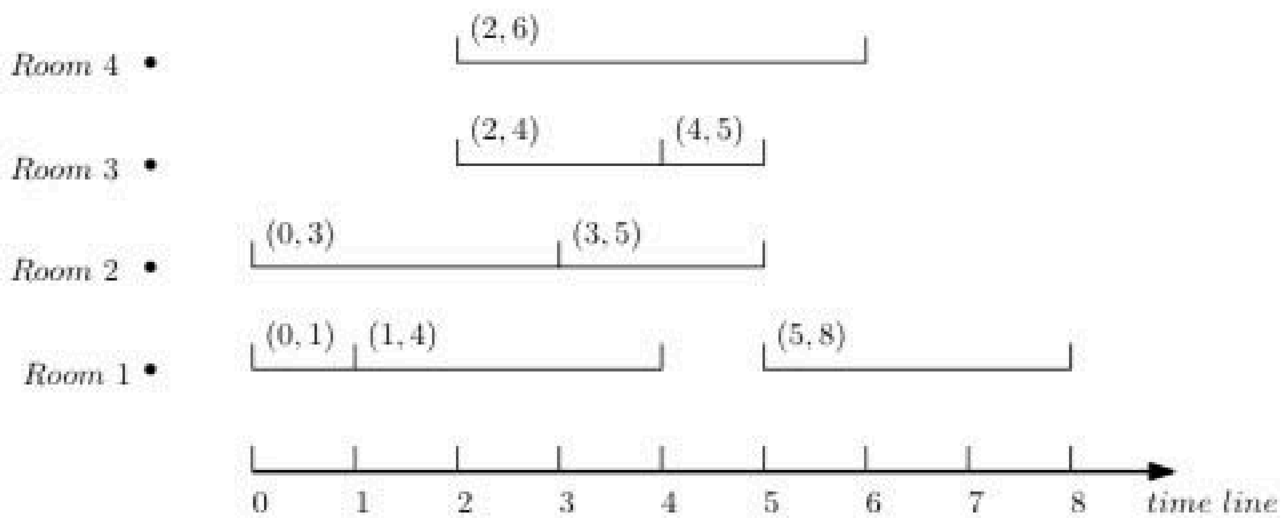


Figure 1: One possible interval partition.

Step 3. The minimum number of classrooms is 4. Note that the Figure is not the unique but the minimum number of classrooms is unique.

(c) The **depth** of a set of intervals is the maximum number of overlapped intervals (lectures) during the whole time line. Hence, the depth is 4

Question 2. (15 points)

(a) (7 points) Find the minimum spanning tree for the graph in Figure 2 using Kruskal's algorithm.

(b) (8 points) Find the minimum spanning tree for the graph in Figure 2 using Prim's algorithm.

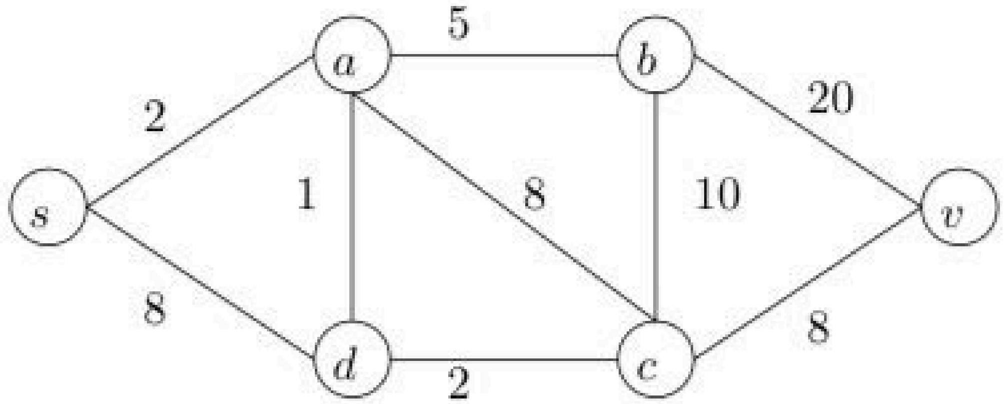


Figure 2: Graph for Question 2.

Solution 2

(a) Kruskal's algorithm:

Step 1. Sort all edges in non-decreasing order by their weights:
$(a, d, 1), (s, a, 2), (d, c, 2), (a, b, 5), (s, d, 8), (a, c, 8), (c, v, 8), (b, c, 10), (b, v, 20)$.

Step 2. Select $(a, d, 1)$.
$A = \{ (a, d) \}$.

Step 3. Select $(s, a, 2)$.
$A = \{ (a, d), (s, a) \}$.

Step 4. Select $(d, c, 2)$.
$A = \{ (a, d), (s, a), (d, c) \}$.

Step 5. Select $(a, b, 5)$.
$A = \{ (a, d), (s, a), (d, c), (a, b) \}$.

Step 6. Don't select $(s, d, 8)$ since vertex s and d are in the same connected component.
$A = \{ (a, d), (s, a), (d, c), (a, b) \}$.

Step 7. Don't select $(a, c, 8)$ since vertex a and c are in the same connected component.
$A = \{ (a, d), (s, a), (d, c), (a, b) \}$.

Step 8. Select $(c, v, 8)$.
$A = \{ (a, d), (s, a), (d, c), (a, b), (c, v) \}$.

Step 9. Don't select $(b, c, 10)$ since vertex b and c are in the same connected component.
$A = \{ (a, d), (s, a), (d, c), (a, b), (c, v) \}$.

Step 10. Don't select $(b, v, 20)$ since vertex b and v are in the same connected component.
$A = \{ (a, d), (s, a), (d, c), (a, b), (c, v) \}$.

Step 11. All edges have been visited and therefore Kruskal's algorithm end.

(b) Prim's algorithm:

Step 1. Start from an arbitrary vertex, e.g., s:
$A = \emptyset$.
Q is the following table. Q can also be written as a set of tuples, i.e., $Q = \{ (node, key, parent) \}$.

Node	s	a	b	c	d	v
Key	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
Parent	NIL	NIL	NIL	NIL	NIL	NIL

Step 2. Choose the vertex s and update key values:
$A = \emptyset$.

Node	s	a	b	c	d	v
Key	0	2	$+\infty$	$+\infty$	8	$+\infty$
Parent	NIL	s	NIL	NIL	s	NIL

Step 3. Choose the vertex a and update key values:
$A = \{ (s, a) \}$.

Node	s	a	b	c	d	v
Key	0	2	5	8	1	$+\infty$
Parent	NIL	s	a	a	a	NIL

Step 4. Choose the vertex d and update key values:
$A = \{ (s, a), (d, a) \}$.

Node	s	a	b	c	d	v
Key	0	2	5	2	1	$+\infty$
Parent	NIL	s	a	d	a	NIL

Step 5. Choose the vertex c and update key values:
$A = \{ (s, a), (d, a), (c, d) \}$.

Node	s	a	b	c	d	v
Key	0	2	5	2	1	8
Parent	NIL	s	a	d	a	c

Step 6. Choose the vertex b and update key values:
$A = \{ (s, a), (d, a), (c, d), (b, a) \}$.

Node	s	a	b	c	d	v
Key	0	2	5	2	1	8
Parent	NIL	s	a	d	a	c

Step 7. Choose the vertex v and update key values:
$A = \{ (s, a), (d, a), (c, d), (b, a), (v, c) \}$.

Node	s	a	b	c	d	v
Key	0	2	5	2	1	8
Parent	NIL	s	a	d	a	c

Step 8. $Q = \emptyset$ and therefore Prim's algorithm ends.

Question 3. (20 points)

(a) (15 points) Use Dijkstra's algorithm to compute a shortest path from s to v in Figure 3. You should give main steps.

(b) (5 points) Does Dijkstra's algorithm work for the case, where some edges can have negative weights? Why?

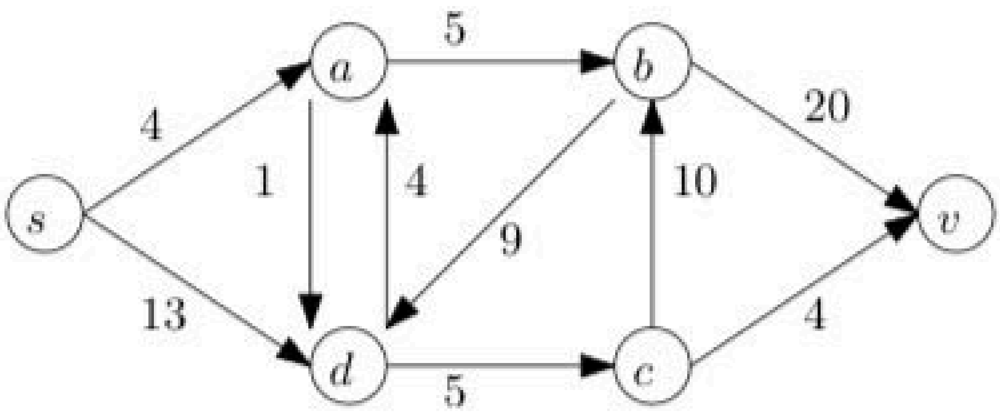


Figure 3: Graph for Question 3

Solution 3

(a) • Dijkstra's algorithm:

Step 1. Start from the source s and maintain 3 variables.
$S = \emptyset$.

Node	s	a	b	c	d	v
distance	0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$
π	NIL	NIL	NIL	NIL	NIL	NIL

Step 2. Add s to set S and update distances.
$S = \{ s \}$.

Node	s	a	b	c	d	v
distance	0	4	$+\infty$	$+\infty$	13	$+\infty$
π	NIL	s	NIL	NIL	s	NIL

Step 3. Add a to set S and update distances.
$S = \{ s, a \}$.

Node	s	a	b	c	d	v
distance	0	4	9	$+\infty$	5	$+\infty$
π	NIL	s	a	NIL	a	NIL

Step 4. Add d to set S and update distances.
$S = \{ s, a, d \}$.

Node	s	a	b	c	d	v
distance	0	4	9	10	5	$+\infty$
π	NIL	s	a	d	a	NIL

Step 5. Add b to set S and update distances.
$S = \{ s, a, d, b \}$.

Node	s	a	b	c	d	v
distance	0	4	9	10	5	29
π	NIL	s	a	d	a	b

Step 6. Add c to set S and update distances.
$S = \{ s, a, d, b, c \}$.

Node	s	a	b	c	d	v
distance	0	4	9	10	5	14
π	NIL	s	a	d	a	c

Step 7. Add v to set S.
$S = \{ s, a, d, b, c, v \}$.

Node	s	a	b	c	d	v
distance	0	4	9	10	5	14
π	NIL	s	a	d	a	c

Step 8. Set S contains all vertices in the given graph and therefore Dijkstra's algorithm ends.

Question 4 (15 points)

(a) (10 points) For the list: 2, 1, 5, 8, 9, 10, 4, 7, 6, 13, 14, and 11. Suppose we have sorted the two halves as list1: 1, 2, 5, 8, 9, 10; and list2: 4, 6, 7, 11, 13, 14. Calculate the number of inversions with one number in list1 and the other number in list2 using O(n) operations. Immediate steps are required.

Answer: Similar to the slides.

(b) (1 points) Suppose $T(1)=1$, and $T(n)=T(n-1)+n$ for $k= 2,3, 4,$ ..... What is $T(n)$ in terms of big O notation?

Answer: $T(n)=T(n-1)+n=\{T(n-2)+n-1\}+n=...T(1)+2+3+...n=O(n^2)$

(c) (1 points) Suppose $T(1)=1$, and $T(n)=T(n-1)+1$ for $k= 2,3, 4,$ ..... What is $T(n)$ in terms of big O notation?

Answer: $T(n)=T(n-1)+1=\{T(n-2)+1\}+1=...$ $T(1)$ $+1+1+...+1$. // (n-1) 1's

=O(n)

(d) (1 points) Suppose $T(1)=1$, and $T(n)=T(n/2)+1$ for $k= 2,3, 4,$ ..... What is $T(n)$ in terms of big O notation?

Answer: $T(n)=T(n/2)+1=\{T(n/2^2)+1\}+1=T(n/2^3)+1+1+1$

$=T(n/2^k)+k$. When n/2k=1, we have $k=\log_2 n$. Thus,

$T(n)= T(n/2^k)+k=T(1)+\log_2 n=O(\log_2 n)$.

(e) (1 points) Suppose $T(0)=1$, $T(1)=1$, and $T(n)=T(n-2)+\log_2 n$ for $k= 2,3, 4,$ ..... What is $T(n)$ in terms of big O notation?

Answer: $T(n)=T(n-2)+\log_2 n=\{T(n-2-2)+\log 2\ (n-2)\}+\log_2 n=...$

$=T(n-2k)$ $+\log_2 (n-2(k-1))$ $+\log 2\ (n-2)+\log_2 n$. When $n-2k=1$,

$T(n)=T(1)+\log_2(3)+...$ $+\log 2\ (n-2)+\log_2 n<=n\log_2 n$. SO, $T(mn)=O(n \log_2 n)$.

(f) (1 points) Suppose $T(1)=1$, and $T(n)=T(n/3)+1$ for $k= 2,3, 4,$ ..... What is $T(n)$ in terms of big O notation?

Answer: $T(n)=T(n/3)+1=\{T(n/3^2)+1\}+1=...T(n/3^k)+k=T(1)+\log_3 k$.

(When $n/3^k=1$, $k=\log 3\ n$).