Test-3 CSC 575 (7 questions, 14.28 x 7 = 100 points)

Name: NADAVATI · RAMESH.

Question 1

Apply Horspool's algorithm to search for the pattern BAOBAB in the text

BESS_KNEW_ABOUT_BAOBABS

Answer:

Given

pattern BAOBAB

BESS_KNEW_ABOUT_BAOBAS

The shift table for the pattern BAOBAB in a text comprised of english letters, the period, and a space will be

c	A	B	C	D	. . .	O	. . .	Z	.	_
t(c)	1	2	6	6	6	3	6	6	6	6

The actual search will proceed as shown below:

```
BESS_KNEW_ABOUT_BAOBABS
                BAOBAB
BAOBAB
                  BAOBAB
        BAOBAB
          BAOBAB
```

Question 2

For the input 30, 20, 56, 75, 31, 19 and hash function $h(K) = K \bmod 11$

a. construct the closed hash table.

Answer:

Given data:-

The list of the keys: 30, 20, 56, 75, 31, 19

The hash function: $h(k) = k \bmod 11$

The hash address

K	30	20	56	75	31	19
h(k)	8	9	1	9	9	8

0	1	2	3	4	5	6	7	8	9	10
								30		
								30	20	
	56							30	20	
	56							30	20	75
31	56							30	20	75
31	56	19						30	20	75

The closed hash table is drawn above and mentioned list of the keys.

Question 3

Solve the instance 5, 1, 2, 10, 6 of the coin-row problem.

Help:

$F(n) = \max\{c_n + F(n-2),\ F(n-1)\}$ for $n > 1$,

$F(0) = 0,\ F(1) = c_1$

Answer:

Given instance 5, 1, 2, 10, 6, use the table generated by the dynamic programming algorithm in solving the problem's instance.

The application of the dynamic programming algorithm to the input 5, 1, 2, 10, 6 yielded the following table.

~~Index 0 1 2 3 4 5 6~~

~~5 1 2 10 6 2~~

Index	0	1	2	3	4	5	6
C		5	1	2	10	6	2
F	0	5	5	7	15	15	17

Using the data in the first six columns, we conclude that the largest amount of money that can be obtained for the input 5, 1, 2, 10, 6 is $F(5) = 15$, which is obtained by taking coins $c_4 = 10$ and $c_1 = 5$.

Question 4

a. Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem:

item	weight	value
1	3	\$25
2	2	\$20
3	1	\$15
4	4	\$40
5	5	\$50

, capacity $W = 6$.

Help:

Consider instance defined by first *i* items and capacity *j* ($j \le W$).

Let $V[i,j]$ be optimal value of such instance. Then

$$V[i,j] = \begin{cases} \max\{V[i-1,j],\ v_i + V[i-1,j-w_i]\} & \text{if } j-w_i \ge 0 \\ V[i-1,j] & \text{if } j-w_i < 0 \end{cases}$$

Initial conditions: $V[0,j] = 0$ and $V[i,0] = 0$

Answer:

let us consider items as "i" and capacity $j (j \le W)$

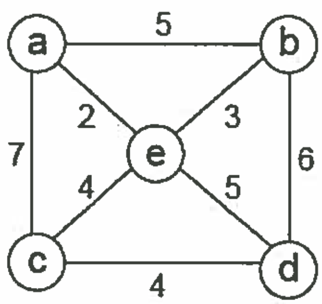
~~$w_1 = 3, v_1$~~

	i	0	1	2	3	4	5	6
	0	0	0	0	0	0	0	0
$w_1 = 3,\ v_1 = 25$	1	0	0	0	25	25	25	25
$w_2 = 2,\ v_2 = 20$	2	0	0	20	25	25	45	45
$w_3 = 1,\ v_3 = 15$	3	0	15	20	35	40	45	60
$w_4 = 4,\ v_4 = 40$	4	0	15	20	35	40	55	60
$w_5 = 5,\ v_5 = 50$	5	0	15	20	35	40	55	65

The maximum value of a feasible subset is $F(5,6) = 65$. The optimal subset is {item 3, item 5}.

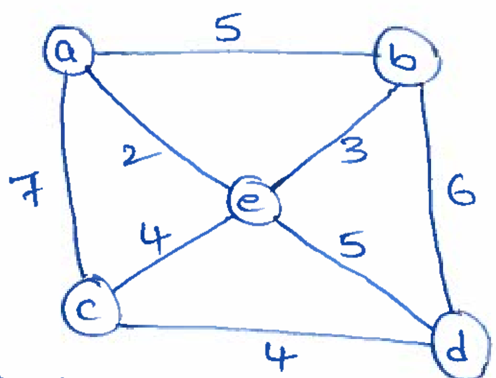
Question 5

Apply Prim's algorithm to the following graph.



Answer:

Given data:



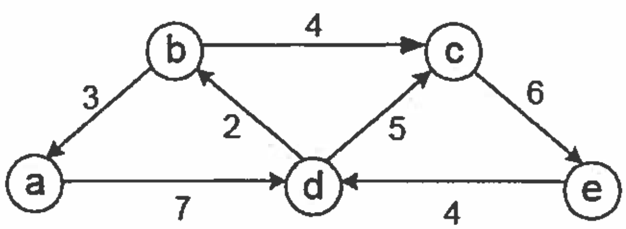
Applying prim's algorithm to the above graph.

Tree vertices	priority queue of remaining vertices
a(-,-)	b(a,5) c(a,7) d(a,∞) e(a,2)
e(a,2)	b(e,3) c(e,4) d(e,5)
b(e,3)	c(e,4) d(e,5)
c(e,4)	d(c,4)
d(c,4)	

The remaining spanning tree found by the algorithm comprises the edges ae, eb, ec, and cd.

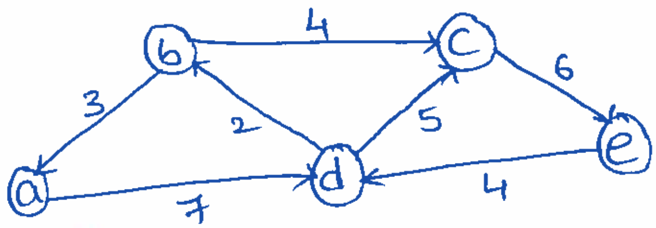
Question 6

Solve the following instances of the single-source shortest-paths problem with vertex a as the source:



Answer:

Trace the algorithm on the given graphs and the instances of the single-source shortest-paths.



Tree vertices	Remaining vertices
a(−,0)	b(−,∞) c(−,∞) d(a,7) e(−,∞
d(a,7)	b(d,7+2) c(d,7+5) e(−,∞)
b(d,9)	c(d,12) e(−,∞)
c(d,12)	e(c,12+6)
e(c,18)	

The shortest paths (identified by following non-numeric labels backwards from a destination vertex to the source) and their lengths are:

From a to d : a — d of length 7

from a to b : a-d-b of length 9

from a to c : a-d-c of length 12

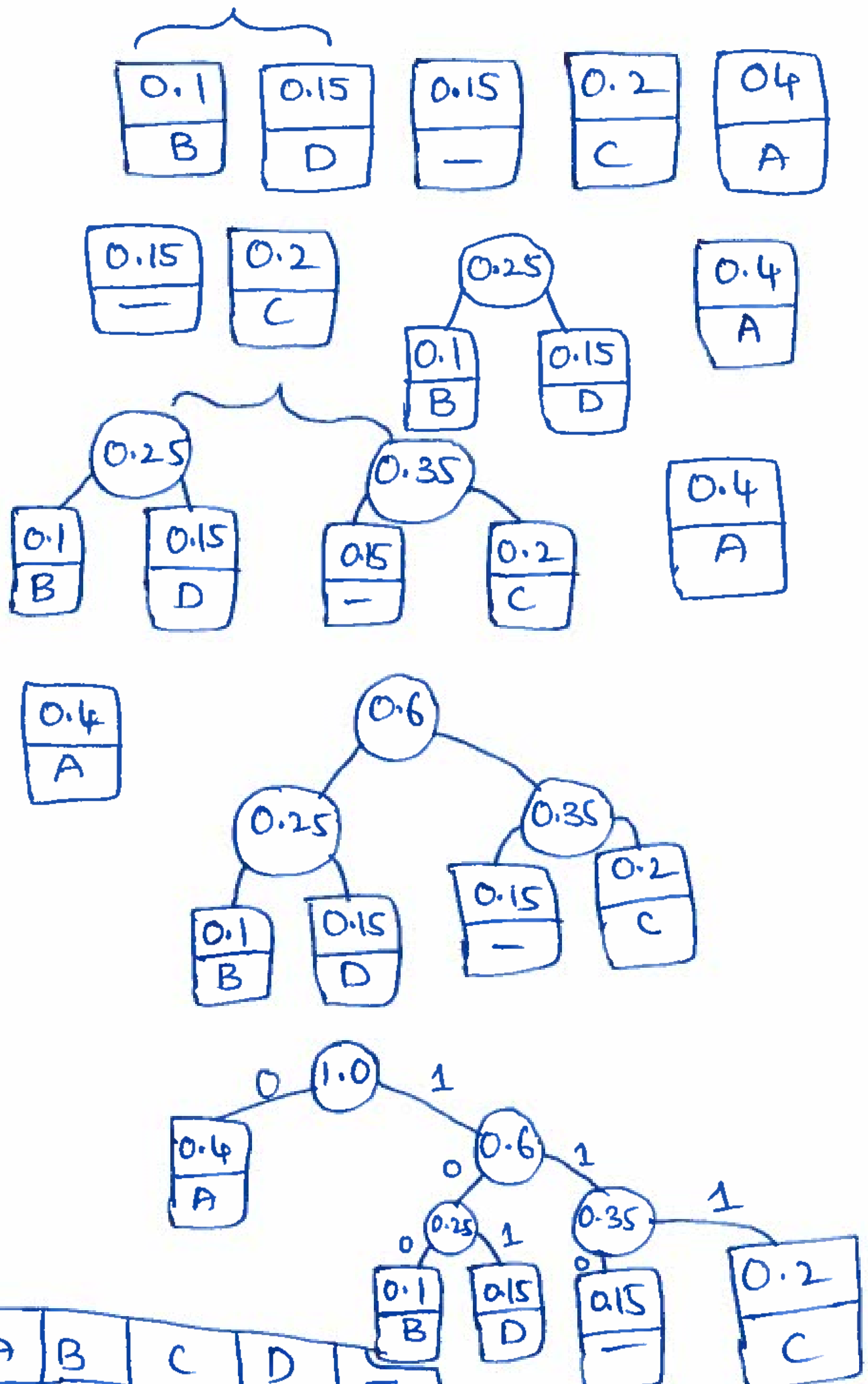
from a to e : a-d-c-e of length 15

Question 7

Construct a Huffman code for the following data:

character	A	B	C	D	_
probability	0.4	0.1	0.2	0.15	0.15

Answer:



character	A	B	C	D	_
probability	0.4	0.1	0.2	0.15	0.15
code word	0	100	111	101	110