School of Comp.
Sci. & Eng.

Homework No.2

April. 13, 2023

Notice

1. 문제는 특별한 이유가 없는 한, 손으로 풀어서 작성한다.
2. 작성된 리포트는 스캔 혹은 사진을 찍어서 하나의 압축된 파일로 묶은 후 PLATO 과제 제출란에 제출하거나 연구실 앞 과제 제출함에 제출한다.
   연구실 : 자연대연구실험동 (건물번호 313동) 313호
3. Due Date : 4월 27일 24시

학과 : 정보컴퓨터공학부

학번 : 201924437

이름 : 김윤하

1. How many different binary search trees can be constructed using six distinct keys? **(Text Book Exercises 3.5 No: 21)**

서로 다른 binary search trees의 개수

$$= \frac{(2n)!}{(n+1)!\, n!}$$

$$n = (key) = 6. \Rightarrow \frac{12!}{(6+1)!\, 6!} = \frac{12!}{7!\, 6!}$$

$$= \frac{\cancel{12} \cdot 11 \cdot \cancel{10} \cdot \overset{3}{\cancel{9}} \cdot \overset{4}{\cancel{8}}}{\cancel{6} \cdot \cancel{5} \cdot \underset{2}{\cancel{4}} \cdot \cancel{3} \cdot \cancel{2}} = 132$$

$$\boxed{\therefore 132}$$

2. Find an optimal circuit for the weighted, direct graph represented by the following matrix W. Show the actions step by step.

$$
W = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{bmatrix} 0 & 8 & 13 & 18 & 20 \\ 3 & 0 & 7 & 8 & 10 \\ 4 & 11 & 0 & 10 & 7 \\ 6 & 6 & 7 & 0 & 11 \\ 10 & 6 & 2 & 1 & 0 \end{bmatrix}
\end{array}
$$

(Text Book Exercises 3.6 No: 28)

Node를 a, b, c, d, e 라 하고, a 노드를 기준으로 거리를 계산해보자.

i) $x \to a$

cost[b][a] = 3
cost[c][a] = 4
cost[d][a] = 6
cost[e][a] = 10

ii) $x \to x' \to a$

cost[b][c][a] = 7+4 = 11
cost[b][d][a] = 8+6 = 14
cost[b][e][a] = 10+10 = 20

cost[c][b][a] = 11+3 = 14
cost[c][d][a] = 10+6 = 16
cost[c][e][a] = 7+10 = 17

cost[d][b][a] = 6+3 = 9
cost[d][c][a] = 7+4 = 11
cost[d][e][a] = 11+10 = 21

cost[e][b][a] = 6+3 = 9
cost[e][c][a] = 2+4 = 6
cost[e][d][a] = 1+6 = 7

iii) $x \to x' \to x'' \to a$

cost[b][(c,d)][a] = min(7+16, 8+11) = 19
cost[b][(c,e)][a] = min(7+17, 10+6) = 16
cost[b][(d,e)][a] = min(8+21, 10+7) = 17

cost[c][(b,d)][a] = min(11+14, 10+9) = 19
cost[c][(b,e)][a] = min(11+20, 7+9) = 16
cost[c][(d,e)][a] = min(10+21, 7+7) = 14

cost[d][(b,c)][a] = min(6+11, 7+14) = 17
cost[d][(b,e)][a] = min(6+20, 11+9) = 20
cost[d][(c,e)][a] = min(7+17, 11+6) = 17

cost[e][(b,c)][a] = min(6+11, 2+14) = 16
cost[e][(b,d)][a] = min(6+14, 1+9) = 10
cost[e][(c,d)][a] = min(2+16, 1+11) = 12

iv) $X \to X' \to X'' \to X''' \to a$

$cost[b][(c,d,e)][a] = \min(7+14,\ 8+17,\ 10+12) = 21.$

$cost[c][(b,d,e)][a] = \min(11+17,\ 10+20,\ 7+10) = 17.$

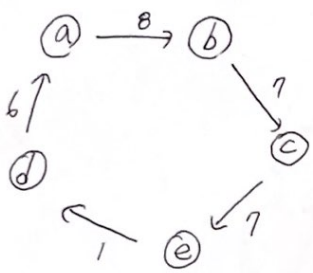
$cost[d][(b,c,e)][a] = \min(6+16,\ 7+16,\ 11+16) = 22.$

$cost[e][(b,c,d)][a] = \min(6+19,\ 2+19,\ 1+17) = 18.$

v) $a \to x \to x' \to x'' \to x''' \to x'''' \to a$

$cost[a][(b,c,d,e)][a] = \min(8+21,\ 13+17,\ 18+22,\ 20+18) = 29$

$\Rightarrow a \to b \to c \to e \to d \to a.$



$= 29.$

3. Assuming a penalty of 1 for a mismatch and a penalty of 2 for a gap, use the dynamic programming algorithm to find an optimal alignment of the following sequences:

$$CCGGGTTACCA$$
$$CGAGTTCA$$

(Text Book Exercises 3.7 No: 33)

	0	C	C	G	G	G	T	T	A	C	C	A
0	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22
C	-2	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
G	-4	-2	-1	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-6	-4	-3	-2	-3	-5	-7	-9	-10	-12	-14	-16
G	-8	-6	-5	-3	-2	-3	-5	-7	-9	-11	-13	-15
T	-10	-8	-7	-5	-4	-3	-3	-5	-7	-9	-11	-13
T	-12	-10	-9	-7	-6	-5	-3	-3	-5	-7	-9	-11
C	-14	-12	-10	-9	-8	-7	-5	-4	-4	-5	-7	-9
A	-16	-14	-12	-11	-10	-9	-7	-6	-4	-5	-6	-7.

CCGG-GTTACCA
-C-GAGTT--CA → Alignment score = 7.

$\boxed{\therefore \text{CGGTTCA}}$

4. Consider the following array:

	1	2	3	4	5	6
1	0	∞	72	50	90	35
2	∞	0	71	70	73	75
3	72	71	0	∞	77	90
4	50	70	∞	0	60	40
5	90	73	77	60	0	80
6	35	75	90	40	80	0

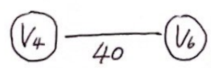
(a) Starting with vertex $v4$, trace through Prim's algorithm to find a mini-mum spanning tree for the graph represented by the array shown here.

(b) Show the set of edges that comprise the minimum spanning tree.

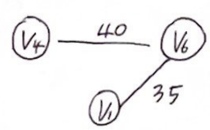
(c) What is the cost of the minimum spanning tree?
(Text Book Exercises 4.1 No: 3)

(a) Prim's minimum spanning tree

i) distance (V_4, V_6) = 40. (min)

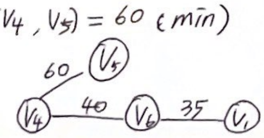


ii) distance (V_6, V_1) = 35 (min)

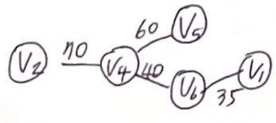


iii) distance (V_4, V_1) = 50 (min) → exist.

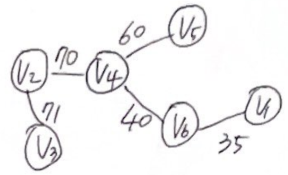
∃ distance (V_4, V_5) = 60 (min)



iv) distance (V_4, V_2) = 70 (min)

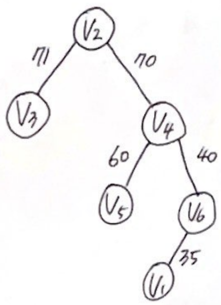


v) distance (V_2 , V_3) = 71 (min)



⇒ minimum spanning tree

(b)



; edges in minimum spanning tree

$\{(V_1, V_6), (V_2, V_3), (V_4, V_5), (V_2, V_4), (V_4, V_6)\}$

(c) cost of the minimum spanning tree

$35 + 40 + 60 + 70 + 71 = \boxed{276}$

5. Modify Dijkstra's algorithm (Algorithm 4.3) so that it computes the lengths of the shortest paths. Analyze the modified algorithm and show the results using order notation. **(Text Book Exercises 4.2 No: 15)**

```
void dijkstra (int n, const number W[][], set_of_edges& F)
{
  index i, vnear;
  edge e;
  index touch[2..n];
  number length[2..n];

  F = Ø;
  for (i = 2; i <= n; i++){                // For all vertices, initialize v1
     touch[i] = 1;                         // to be the last vertex on the
     length[i] = W[1][i];                  // current shortest path from
  }                                        // v1, and initialize length of
                                           // that path to be the weight
                                           // on the edge from v1.

 repeat (n - 1 times){                     // Add all n - 1 vertices to Y.
    min = ∞;
    for (i = 2; i <= n; i++)               // Check each vertex for
       if (0 ≤ length[i] < min){           // having shortest path.
          min = length[i];
          vnear = i;
       }
    e = edge from vertex indexed by touch[vnear]
        to vertex indexed by vnear;
    add e to F;
    for (i = 2; i <=n; i++)
       if (length[enear] + W[vnear][i] < length[i]){
          length[i] = length[vnear] + W[enear][i];
          touch[i] = enear;                // For each vertex not in Y,
       }                                   // update its shortest path.
    length[vnear] = -1;                    // Add vertex indexed by vnear
 }                                         // to Y.
}
```

① 출발 - 다른 노드 까지 거리 ∞ Initialization

② 출발 노드 기점 탐색 시작

☆ ③ 현재 노드와 인접한 모든 노드에 대해, 기존 거리보다 현재 노드 통한 거리 짧으면 Update.

④ 방문하지 않은 정점 중 거리가 가장 짧은 노드를 다음 방문.

⑤ ③.④ 반복.

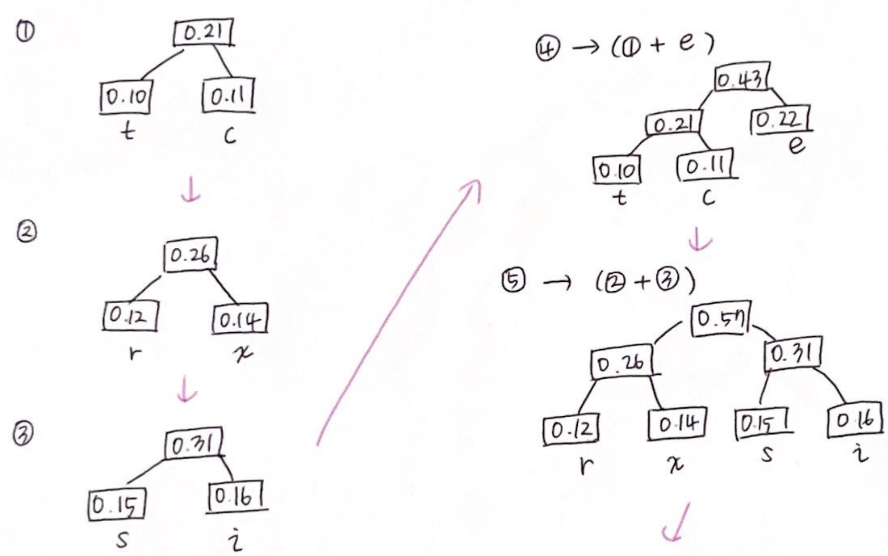
시간 복잡도 → $O(n^2)$

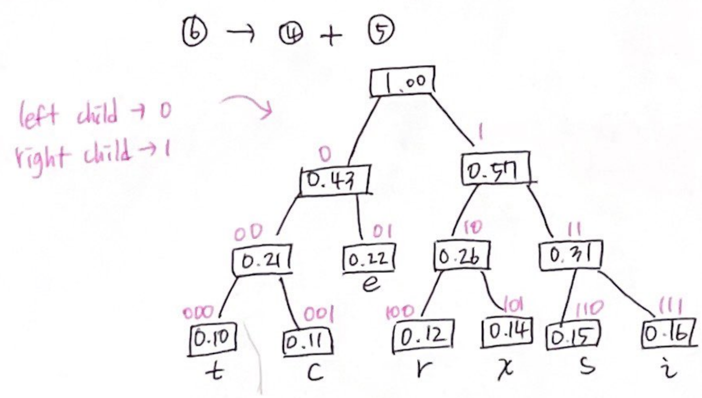
6. Use Huffman's algorithm to construct an optimal binary prefix code for the letters in the following table.
(Text Book Exercises 4.4 No: 27)

Letter	:	c	e	i	r	s	t	x
Probability	:	0.11	0.22	0.16	0.12	0.15	0.10	0.14

작은 값부터 순서대로 나열하면,

$\underbrace{t\,(0.10) < c\,(0.11)}_{①} < \underbrace{r\,(0.12) < x\,(0.14)}_{②} < \underbrace{s\,(0.15) < i\,(0.16)}_{③} < e\,(0.22)$





⑦ e → 01
t → 000
c → 001
r → 100
x → 101
s → 110
i → 111

7. Decode each bit string using the binary code in Exercise 26. →

(a) 01100010101010

(b) 1000100001010

(c) 11100100111101

(d) 1000010011100

(Text Book Exercises 4.4 No: 28)

A : 00
B : 011
I : 10
M : 111
S : 110
X : 0101
Z : 0100

(a) 011 00 0101 0101 010...
B A X X .. ; BAXX

(b) 10 00 10 00 0101 010...
I A I A X .. ; IAIAX

(c) 111 00 10 011 110 1...
M A I B S .. ; MAIBS

(d) 10 00 0100 111 00
I A Z M A ; IAZMA