소프트웨어학부 20202003002 박상훈

C17

4.

a)

$r^2 = r + 6$

$r^2 - r - 6 = 0$

$(r-3)(r+2) = 0$

$r = 3$ or $r = -2$

$a_n = \alpha_1 3^n + \alpha_2 (-2)^n$

$3 = a_0 = \alpha_1 + \alpha_2$

$6 = a_1 = 3\alpha_1 - 2\alpha_2$

$\alpha_1 = \frac{12}{5}, \ \alpha_2 = \frac{3}{5}$

$\therefore a_n = \frac{12}{5} 3^n + \frac{3}{5} \cdot (-2)^n$

(b)

$r^2 = 7r - 10$

$r^2 - 7r + 10 = 0$

$(r-5)(r-2) = 0$

$r = 5$ or 2

$a_n = \alpha_1 2^n + \alpha_2 5^n$

$2 = a_0 = \alpha_1 + \alpha_2$

$1 = a_1 = 2\alpha_1 + 5\alpha_2$

$\alpha_1 = 3, \alpha_2 = -1$

$a_n = 3 \cdot 2^n - 5^n$

(c)

$r^2 = 6r - 8$

$r^2 - 6r + 8 = 0$

$(r-4)(r-2) = 0$

$r = 4$ or 2

$a_n = \alpha_1 2^n + \alpha_2 4^n$

$4 = a_0 = \alpha_1 + \alpha_2$

$10 = a_1 = 2\alpha_1 + 4\alpha_2$

$\alpha_1 = 3, \ \alpha_2 = 1$

$a_n = 3 \cdot 2^n + 4^n$

(d)

$r^2 = 2r - 1$

$r^2 - 2r + 1 = 0$

$(r-1)^2 = 0$

$r = 1$

$a_n = \alpha_1 + \alpha_2 n$

$4 = a_0 = \alpha_1$

$1 = a_1 = \alpha_1 + \alpha_2$

$\alpha_1 = 4, \ \alpha_2 = -3$

$a_n = 4 - 3n$

(e)

$r^2 = 1$

$r^2 - 1 = 0$

$(r-1)(r+1) = 0$

$r = -1$ or 1

$a_n = \alpha_1 + \alpha_2 (-1)^n$

$5 = a_0 = \alpha_1 + \alpha_2$

$-1 = a_1 = \alpha_1 - \alpha_2$

$\alpha_1 = 2, \ \alpha_2 = 3$

$a_n = 2 + 3(-1)^n$

(f)

$r^2 = -6r - 9$

$r^2 + 6r + 9 = 0$

$(r+3)^2 = 0$

$r = -3$

$a_n = \alpha_1 (-3)^n + \alpha_2 n (-3)^n$

$3 = a_0 = \alpha_1 + \alpha_2$

$-3 = a_1 = -3\alpha_1 - 3\alpha_2$

$\alpha_1 = 3, \ \alpha_2 = -2$

$a_n = 3 \cdot (-3)^n - 2n(-3)^n$

$= (3 - 2n)(-3)^n$

(g)

$r^2 = -4r + 5$

$r^2 + 4r - 5 = 0$

$(r+5)(r-1) = 0$

$r = -5$ or 1

$a_n = \alpha_1 + \alpha_2 (-5)^n$

$2 = a_0 = \alpha_1 + \alpha_2$

$8 = a_1 = \alpha_1 - 5\alpha_2$

$\alpha_1 = 3, \ \alpha_2 = -1$

$a_n = 3 - (-5)^n$

24.

a)

$a_n = n2^n \Rightarrow a_{n-1} = (n-1)2^{n-1}$

$2a_{n-1} + 2^n = 2((n-1)2^{n-1}) + 2^n$

$= (n-1)2^n + (1)2^n$

$= (n-1+1)2^n$

$= n2^n = a_n$

b).

$r = 2 \Rightarrow a_n^{(h)} = d \cdot 2^n$

$a_n = a_n^{(h)} + a_n^{(p)}$

$= d \cdot 2^n + n2^n$

c)

$2 = a_0 = d$

$\therefore a_n = (2+n)2^n$

28.

a)

$r = 2 \Rightarrow a_n^{(h)} = d \cdot 2^n$

$F(n) = 2n^2 = 2n^1 1^n$

$\Rightarrow a_n^{(p)} = (p_2n^2 + p_1n + p_0)1^n$

$= p_2n^2 + p_1n + p_0$

$a_n = 2a_{n-1} + 2n^2$

$p_2n^2 + p_1n + p_0 = 2p_2(n-1)^2 + p_1(n-1) + p_0 + 2n^2$

$\Rightarrow 0 = (-p_2 - 2)n^2 + (-p_1 + 4p_2)n + (-2p_2 + 2p_1 - p_0)$

$-p_2 - 2 = 0 \;/\; -p_1 + 4p_2 = 0 \;/\; -2p_2 + 2p_1 - p_0 = 0$

$p_0 = -12, \; p_1 = -8, \; p_2 = -2$

$a_n^{(p)} = p_2n^2 + p_1n + p_0 = -2n^2 - 8n - 12$

$a_n = a_n^{(h)} + a_n^{(p)}$

$= d \cdot 2^n - 2n^2 - 8n - 12$

(b)

$4 = a_1 = 2d - 2 - 8 - 12$

$26 = 2d$

$d = 13$

$\therefore a_n = d \cdot 2^n - 2n^2 - 8n - 12$

$= 13 \cdot 2^n - 2n^2 - 8n - 12$

2020203002 소프트웨어학부 박상진

C20

Theorm 1

• There is a simple path between any pair of vertices in a connected undirected graph.

Proof.

• u와 v를 connected undirected graph의 vertices 라 가정. 그럼 Graph가 연결되어 있기 때문에 u와 v 사이에 적어도 하나이상의 simple path가 존재.
만약, 하나의 path를 골랐을때 그것이 simple이 아니면 simple 한 path가 존재한다는 것을 의미한다.

Therom 2.

• A connected multigraph has an Euler circit iff each vertex has even degree.

Proof

• Euler circit은 모든 vertex를 한번만 통과하고, 인접한 vertex로 끝이난다.
그럼 Euler circit을 가진 Graph를 통과하는 동안 한 vertex를 통해 들어오고, 다른 vertex를 통해 나간다.
따라서 모든 vertex에서는 짝수의 degree 여야한다.

Theorm 3

• A connected multigraph has an Euler path (but not an Euler circuit) iff it has exactly 2 vertices of odd degree.

Proof

• 가정 2에서 Euler Circit을 추가했다고 가정.
그럼 Euler path에서 모든 vertex에 대해 이동할 수 있으나 같은 vertex에서 끝나지 않는다.
따라서 이것은 Edge가 추가되었다는 것인데, 이것이 두 vertex의 degree를 odd로 만든다.

2020203002 소프트웨어학부 박상진

C의

3.

a) a

b) a, b, c, d, f, h, j, q, t

c) e, g, i, k, l, m, n, o, p, r, s, u

d) q, r

e) c

f) p

g) a, b, f

h) e, f, l, m, n

4.

a) a

b) a, b, d, e, g, h, i, o

c) c, f, j, k, l, m, n, p, q, r, s

d) x

e) d

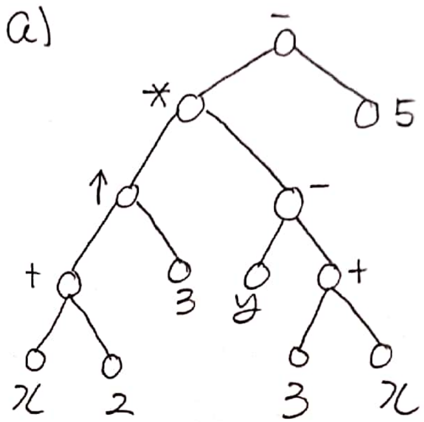
f) p

g) a, b, g

h) e, f, g, j, k, l, m

16

a)



(b)

$- * \uparrow + x\, 2\, 3 - y + 3\, x\, 5$

(c)

$x\, 2 + 3 \uparrow y\, 3\, x + - * 5 -$

(d)

$((((x+2)\uparrow 3) * (y-(3+x))) - 5)$

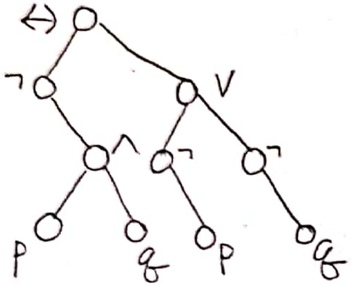
18.

a)

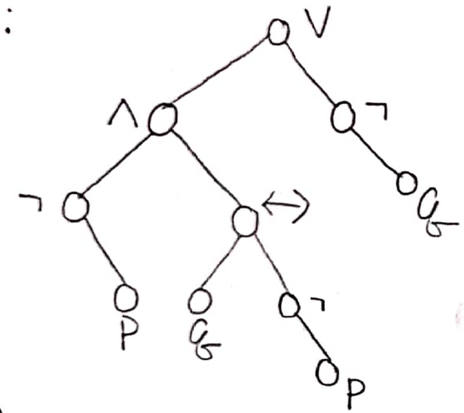
1: ¬(p∧q)↔(¬p∨¬q)

2: (¬p∧(q↔¬p))∨¬q

1:



2:



b)

1: ↔ ¬ ∧ p q ∨ ¬ p ¬ q

2: ∨ ∧ ¬ p ↔ q ¬ p ¬ q

c)

1: p q ∧ ¬ p ¬ q ¬ ∨ ↔

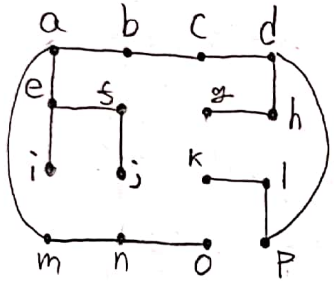
2: p ¬ q p ¬ ↔ ∧ q ¬ ∨

d)

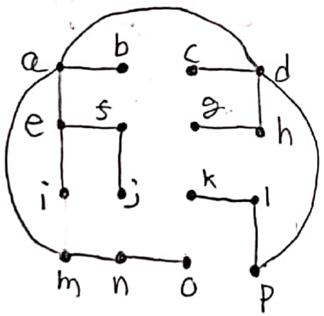
1: ((¬(p∧q))↔((¬p)∨(¬q)))

2: (((¬p)∧(q↔(¬p)))∨(¬q))

(a) prim's algorithm



(b). kruskal's algorithm



18.

G는. 연결된 가중치 그래프, T는 G의 가장 작은 spanning tree이다.

그리고 e는 G에서 가장 가중치가 작은 edge이다.

만약 무게가 w인 e가 T에 포함이 되어있지 않다면, 이것은 T는 가중치가 가장 작은 edge가 적어도 w보다 크다는 것을 의미한다.

만약 T에 e를 포함시킨 새로운 그래프인 T′는 Simple circit을 포함할 것이다. 우리가 만약 Simple circit에서 다른 edge f를 제거하면, 그것을 T″라 하겠다. 그리고 T″는 T보다 가중치가 작을 것이다.

그 이유는 방금 뺀 edge f가 edge e보다 가중치가 크기 때문이다.

그로써 우리는 e가 포함된 T를 얻는데 이것이 가장 가중치가 작아 증명이 완료된다.