Exercici 1

a) El nombre de paquets resultants serà el nombre

Els paquets s'aniran colocant mentres el de sota pugui resistir el s paquets de sobre. Sen aquest pes dels que tenim a sobre - el PM del paquet que mirem

$$Es\_pet = (P_{n+1} + \dots P_k) \cap (P_T \leq PM_n)$$

b) En aquest cas seleccionant el paquet primer el paquet més lleuger tindrem que quan $P > PM$ es pari llavors PM serà el del paquet que mirem en el moment $\max(PM_i (P_{n+1} \dots P_k))$

si $max != PM$ llavors parem dapilar

c) No ja que per exemple el paquet més lleuger també podria ser el que aguantés menys pes i al ficar un a sobre que pesés més que l'anterior no seria optim basicament pel fet de que a sota de tot també s'hauria de mirar que sos el que aguantés més pes. Aixi que podem dir que NO és optim.

J) El criteri que proposaria és començar pel que aguanti més pes i tingui el pes més petit en relació. Seguiria, ficaria el paquet amb el ratió P/PM a sota i aniria apilant.

ex 1/3 o 4/13

1/3 = 0,33 ; 4/13 = 0,3 → seria millor.

Aparta d'això si R tingessin el mateix ratio ficaria el paquet amb més PM a sota.

Exercici 2

a)

```
int Recursio( int N, int direccioN, int[] peces, int[][][][]
combinacio) {
    int compt = 0;
    if( N == M ) {
        return Min( combinacio[N][N-1][direccioN][0], combinacio[N][N-1][direccioN][1] )
    }
    else if( N = 0 ) {
        compt = Recursio(N+1)
        return 0
    }
```

b) Utilitzarem una matriu M x M on hi és el nombre de peces a colocar; bo

```
    else {
        compt = Min(Recursio(N+1, 0, int peces, combinacio),
                    Recursio*(N+1, 0, peces, combinacio*))
    }
return compt;
```

b) Utilitzarem una taula amb 3 paràmetres, 1 per el numero de peça, 1 per la direcció que està i l'altre el numeró de posició en que es coloca.

per poder reconstruir només necesitem una taula amb la peça i la direcció que estigui ordenat de últim a colocar i primer.

Exercici 3

a)

En aquest cas al tenir un graf poc dens, on tenim un numeró semblant de vertex i arestes utilitzarem l'implementació amb cues de prioritat ja que aquesta depen tan del numeró de vèrtex com d'arestes i en aquest cas que no tenim un numeró d'arestes molt gran és l'idoni

exemple:

V = 5

E = 8

si utilitzem cues de prioritat → $8 \cdot \log 5 \simeq 5{,}6$

si s'utilitza un amb matriu d'adjacència → $5^2 = 25$

b)

aquest es el cas contrari, mateixos vèrtex però molts més arestes. En aquest cas implementarem per matriu d'adjacencia ja que depen unicament dels vèrtex que tingui el graf en aquell cas concret:

exemple:

V = 5

E = 200

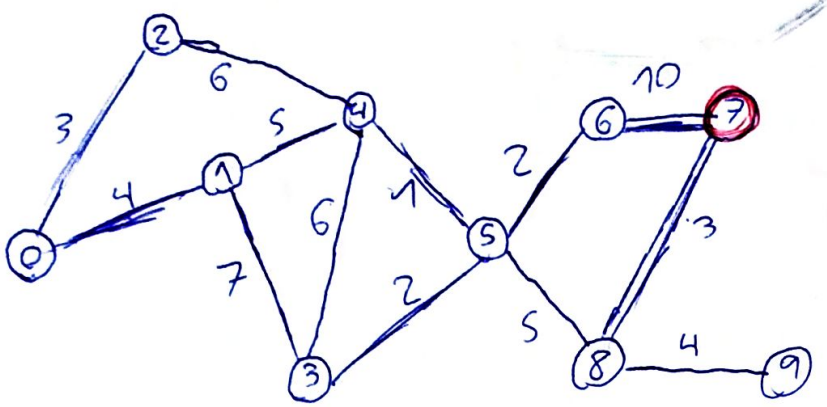
si utilitzem matriu d'adjacència → $5^2 = 25$

si utilitzem cues de prio → $200 \cdot \log 5 \simeq 139{,}8$

Exercici 4

- graf principal
- vèrtex en el MST
- aresta dintre el MST
- possible aresta

Estat inicial:

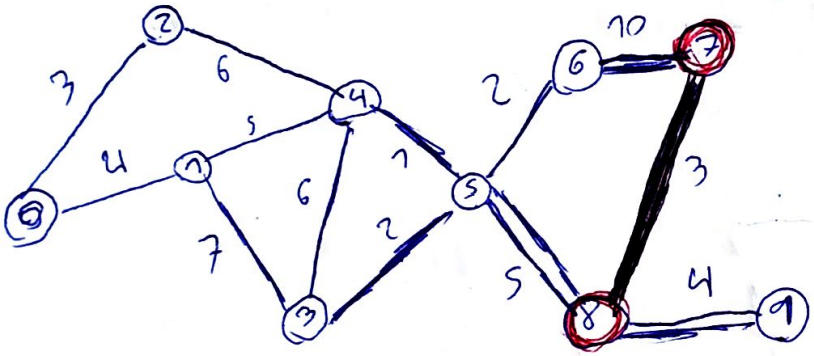


No vist: 0,1,2,3,4,5,9

En el graf: 6,8

En l'arbre MST: 7

1:

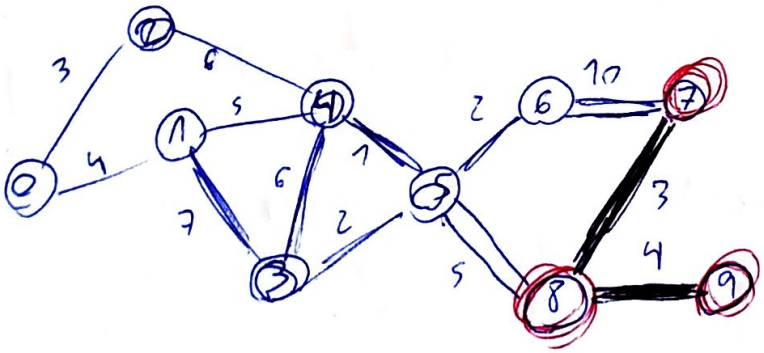


No vist: 0,1,2,3,4,

En el graf: 5,6,9

En el MST: 7,8

2:

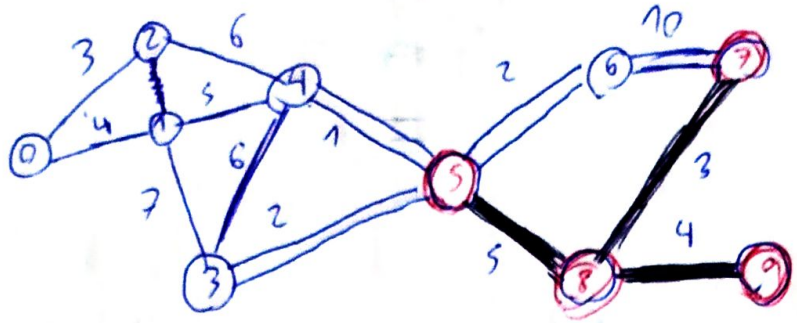


No vist: 0,1,2,3,4

En el graf: 5,6

En el MST: 7,8,9

3.

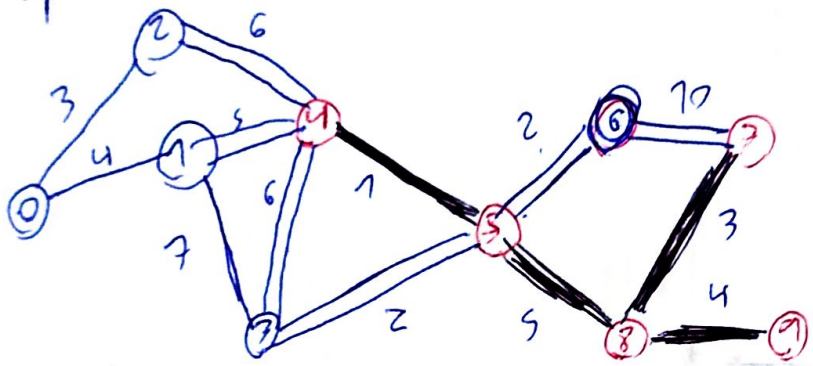


No vist: 0, 1, 2

En el graf: 3, 4, 6

En el MST: 5, 7, 8, 9

4

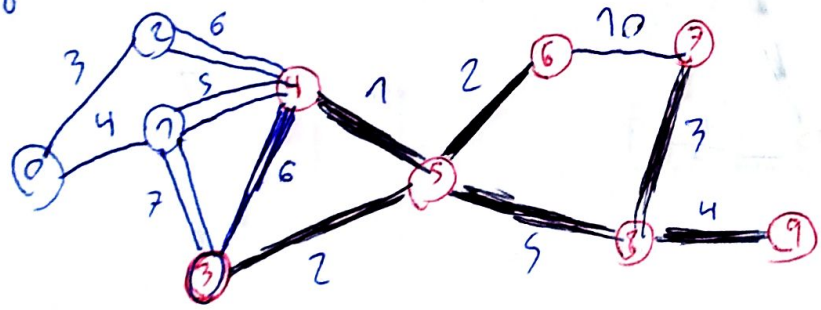


No vist = 0

En el graf: 1, 2, 3, 6

En el MST: 4, 5, 7, 8, 9

6

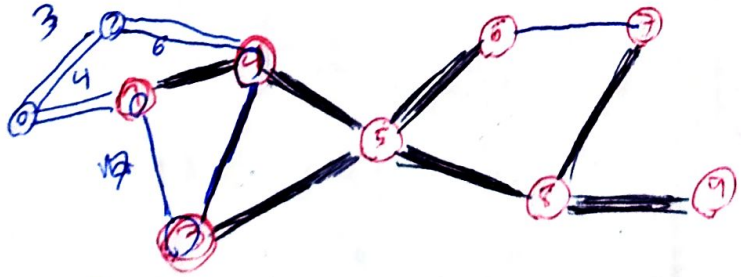


No vis: 0

En el graf: 1, 2

En el MST: 3, 4, 5, 6, 7, 8, 9

7

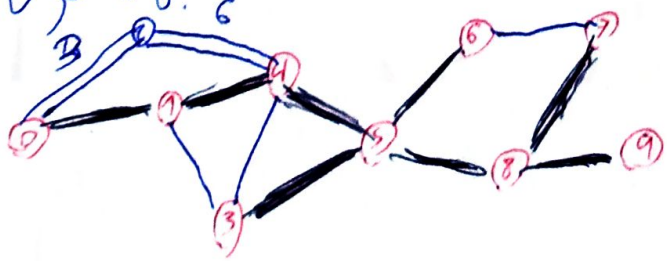


No vist: Null

En el graf: 0, 2

En el MST: ~~2~~, 3, 4, 5, 6, 7, 8, 9

Distancia graf 8.

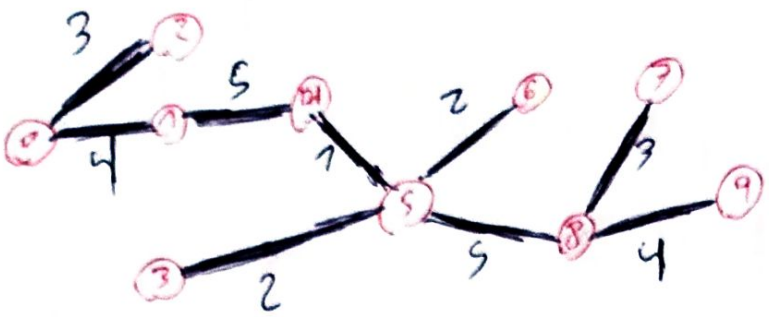


No vist: Null

En el graf: ~~0~~

En el MST: 0, 1, 3, 4, 5, 6, 7, 8

Estat final.



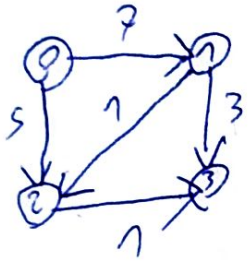
No visit: Null

En el Grafo: Null

En el MST: 0,1,2,3,4,5,6,7,8,9

Exercici 5

Estat inicial



	0	1	2	3
0	0	7	5	∞
1	∞	0	1	3
2	∞	∞	0	1
3	∞	∞	∞	0

n=1

	0	1	2	3
0	0	7	5	∞
1	∞	0	1	
2	∞		0	
3	∞			0

n=2

	0	1	2	3
0	0	7	5	~~10~~
1	∞	0	1	~~3~~
2	∞	∞	0	1
3	∞	∞	0	0

n=3

	0	1	2	3
0	0	7	5	10
1	∞	0	1	3
2	∞	∞	0	1
3	∞	∞	∞	0

n=4

	0	1	2	3
0	0	7	5	10
1	∞	0	1	3
2	∞	~~10~~	0	1
3	∞	∞	∞	0

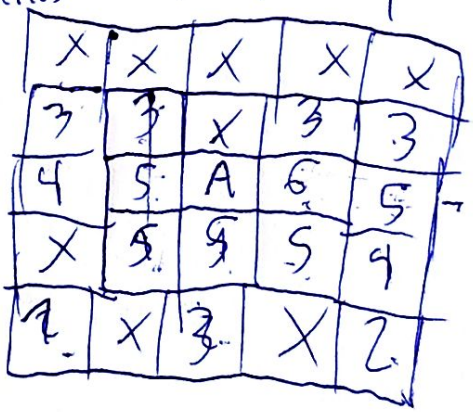
apartat b)

Valdria canvi de pes aqui voldria dir que amb les visibilitats que tenim a n=4 podriem arribar de 3 a 2 o podriem arribar més rapid en el cas de que hi pogessim arribar previament:

Exercici 6

1. Que en cada torn has de realitzar 2 moviments, que hi ha molts més possibilitats per tirar ja que pots escollir moure dos de les 4 reines, en més d'una direcció i més d'una casella a l'hora. per tant en cada elecció del comencia teniem 8 opcions on tirar i amb les amazones com a minim 64 x 8 x ~~caselles~~ costats lliures x passos que podria donar per cada moviment. també haurem d'afegir la dificultat a cada moviment de haver de bloquejar una casella lliure del taulell

2. L'heuristica que vaig fer servir era simplement valorar les caselles a les que tenia accés cada reina. per exemple:



En aquest cas li fico un valor a cada casella que és el numeró de caselles buides que té al voltant. amb aixo faig suma del valor de la casella que pugui arribar l'amazona en un moviment dividit per com esta de lluny per aquest A tindriem que $h(A) = 32/1 + 15/2 = 39$

3. Vaig implementar també una poda Alpha-Beta

4. És pot utilitzar per a guardar els taulers ja calculats i així no haver de repetir si el tornem a trobar. Jo no ho vaig implementar.

5. Rapidessa ja que al principi pot tardar un munt a calcular, cada casella possible i al anar-se reduint redueix drasticament el temps de processament al final. A més com més petit el tauler més petits els valors que calcular i més propers a una heurística realista.