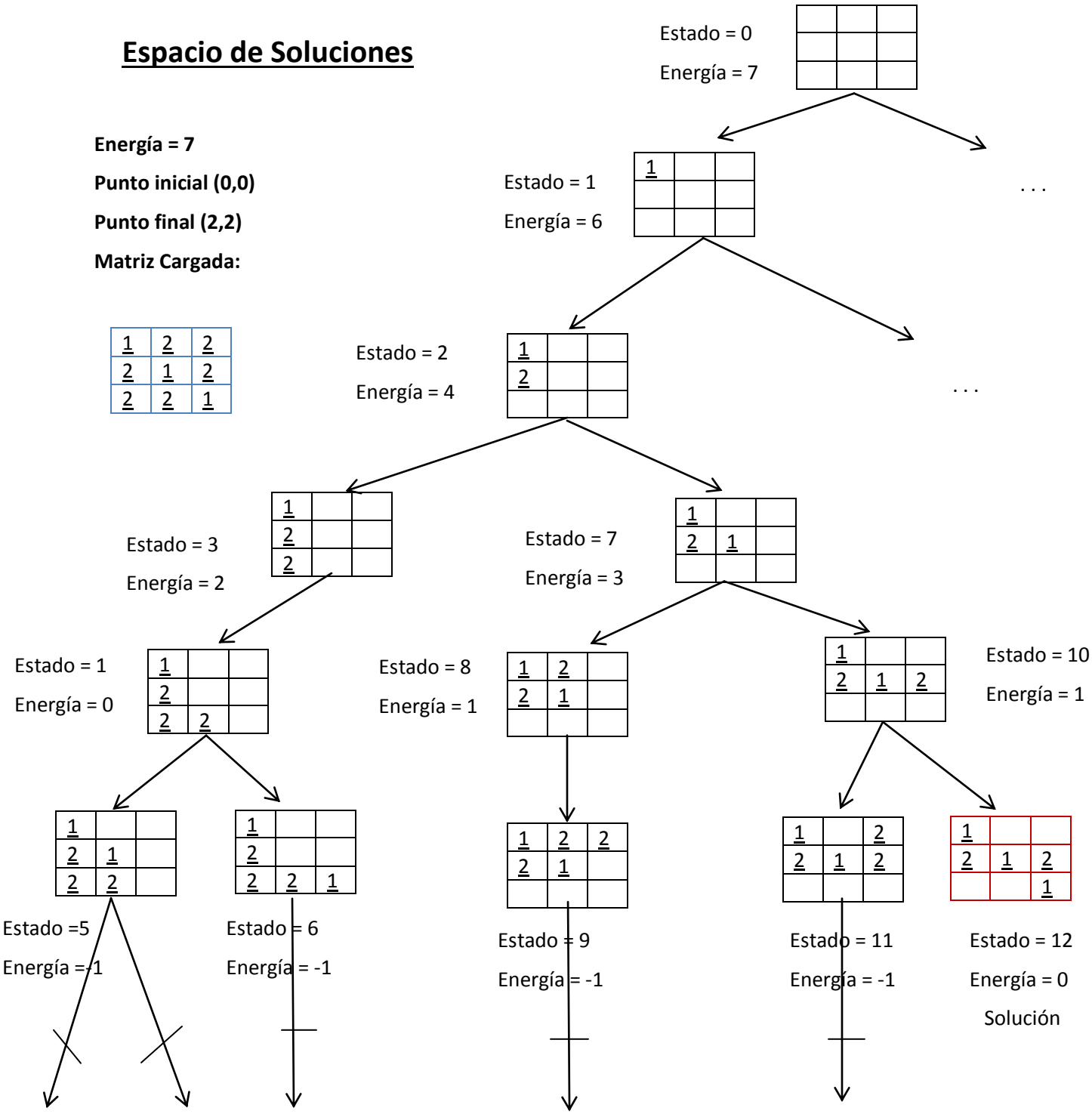
Espacio de Soluciones



Restricciones: **Explícitas: Si la celda no corresponde a la matriz o estoy en una celda ya visitada**

Implícitas: Si la energía es menor a cero

- **La cantidad de arcos se genera a partir de los posibles movimientos que se pueden realizar.**
- **Una vez encontrada la solución, termina**

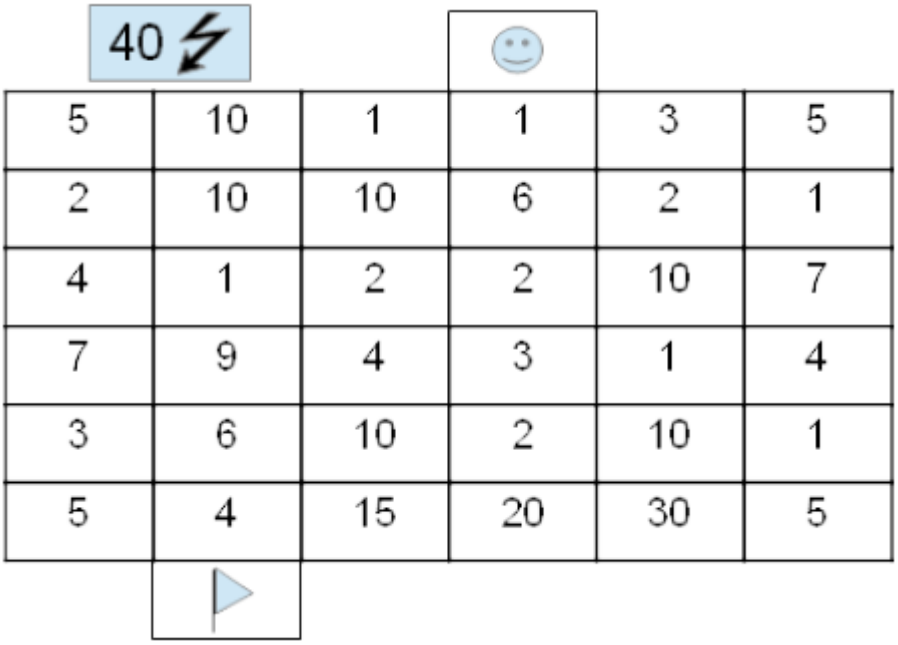
Estrategias de Podas

- Antes de ingresar a una celda, preguntar si la energía disponible menos la energía de la celda es aceptable (mayor a 0).

- Sea E= Energía disponible y C=cantidad de celdas hasta llegar a la celda final. Si E-C es menor a cero podar.

Diferencia con poda y sin poda

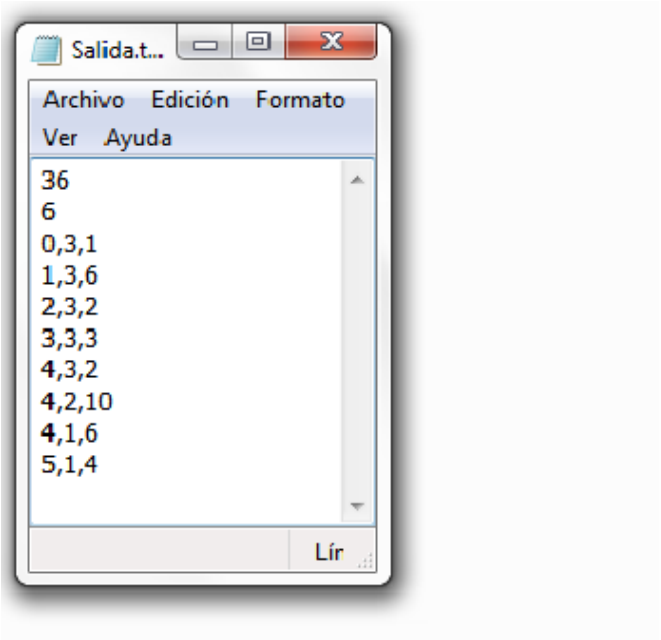
Tomando como Ejemplo dicho tablero:



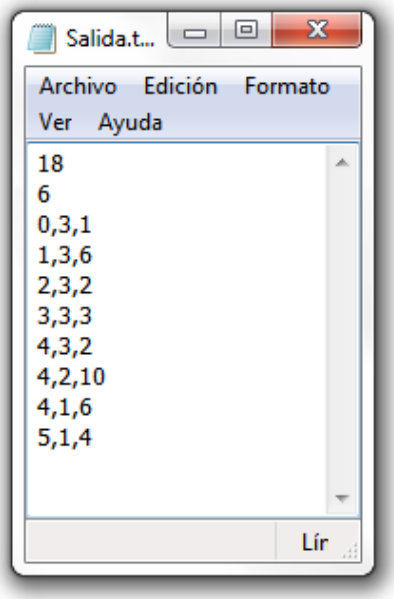
Y la poda:

- Antes de ingresar a una celda, preguntar si la energía disponible menos la energía de la celda es aceptable.

Solución sin poda:



Solución con poda:



Podemos ver que sin la poda recorremos el doble de estados.

- En la implementación uso una lista de puntos, en donde voy guardando el camino por el cual el personaje avanza y lo utilizo para no repetir un casillero ya “visitado”. De esta manera es mas ineficiente ya que para saber si esta visitado es O(n) siendo n la cantidad de puntos registrados hasta el momento.
- La mejor implementación debería ser la de una matriz booleana, logrando acceso directo. O(1).

Evaluación en clase:

Dado el siguiente tablero

5 1 1 10	Punto de partida: (0,2)
7 10 3 2	Punto de llegada: (2,0)
4 1 2 1	Energía: 15

Resultado sin poda:

35
1
0,2,1
1,2,3
1,3,2
2,3,1
2,2,2
2,1,1
2,0,4

Resultado con poda:

13
1
0,2,1
1,2,3
1,3,2
2,3,1
2,2,2
2,1,1
2,0,4