

Trabajo Práctico 3 Diversión NP-Completa

TEORÍA DE ALGORITMOS
(75.29) CURSO BUCHWALD - GENENDER

Nombre	Padrón
Denise Dall'Acqua	108645
Martín Alejo Polese	106808
Nicolás Agustín Riedel	102130

Índice

1. Demostración: NP	4
1.1. Que los barcos no sean adyacentes	4
1.2. Que las demandas de las filas y columnas se cumplan de manera exacta	4
1.3. Que se coloquen todos los barcos	4
2. Demostración: NP-Completo	5
3. Algoritmo de Backtracking	5
4. Ejemplos de ejecución	5
5. Algoritmo de aproximación	5
6. Medición empírica	5
7. Conclusiones	5

Consigna

Resolución

1. Demostración: NP

Para que un problema se encuentre en NP, se debe poder encontrar un validador que valide si la solución es correcta, y lo haga en tiempo polinomial.

Nuestro problema, está dado por:

- Una lista con las demandas para cada fila
- Una lista con las demandas para cada columna
- Una lista de k barcos (donde el barco s tiene b_s de largo)

La solución, está dada por una Matriz de tamaño $n * m$, donde para cada casillero ij :

- Si no hay barco, entonces $\text{Matriz}[i][j] = \text{None}$
- Si hay un barco, entonces $\text{Matriz}[i][j] = s$

Nota: Siendo s el índice de dicho barco

El validador entonces verifica lo siguiente:

- Que los barcos no sean adyacentes
- Que las demandas de las filas se cumplan de manera exacta
- Que las demandas de las columnas se cumplan de manera exacta
- Que se coloquen todos los barcos

1.1. Que los barcos no sean adyacentes

Para verificar que los barcos no sean adyacentes, basta con recorrer cada casillero de la matriz ($n * m$) y en cada celda visitar las 8 celdas vecinas (un cuadrado). En cada una de las celdas vecinas que visito, veo si hay otro barco distinto, y si lo hay, la solución no es válida. Visitar las 8 celdas vecinas se hace en tiempo constante $O(1)$. Por lo tanto, verificar que no haya barcos adyacentes tiene un costo de $O(n * m)$

1.2. Que las demandas de las filas y columnas se cumplan de manera exacta

Para verificar que las demandas de las filas se cumplan, basta con recorrer cada casillero de la matriz ($n * m$) y en cada casillero que haya un barco, restar 1 a la demanda de la fila y columna a la que pertenece. Finalmente, se recorre la lista con las demandas para cada fila y nos fijamos que sea todo igual a cero $O(n)$. Lo mismo con la lista de las demandas para cada columna $O(m)$. Esto nos da un costo total de $O(n * m)$

1.3. Que se coloquen todos los barcos

Para verificar que se colocaron todos los barcos, hay que recorrer cada casillero de la matriz ($n * m$), y en el caso de que en el casillero haya un barco, se agrega a los barcos visitados. Luego se

recorren todos los barcos $O(k)$ y nos fijamos que hayan sido todos colocados. Esto nos da un costo total de $O(n * m)$.

En conclusion, como pudimos validar una solucion al problema en tiempo polinomial, podemos afirmar que el problema se encuentra en NP.

2. Demostración: NP-Completo

3. Algoritmo de Backtracking

4. Ejemplos de ejecución

5. Algoritmo de aproximación

6. Medición empírica

7. Conclusiones