INF-221 Algoritmos y Complejidad, 2020-2

Tarea 1

Profesor: Diego Arroyuelo
Ayudantes: Gabriel Carmona, Martín Crisóstomo, Nicolás Rodríguez
gabriel.carmonat@sansano.usm.cl,
martin.crisostomo@sansano.usm.cl,
nicolas.rodriguezh@sansano.usm.cl.

Fecha de Entrega: 12 de noviembre, 2020
Plazo máximo de entrega: 5 días.

1. Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Puede usar los lenguajes de programación C, C++, Python, y Java.

2. Problema 1: Las Formas de Sumar un Número

Dada un múmero entero t y una lista L de n enteros, queremos encontrar las distintas formas de sumar t usando valores de la lista L. Por ejemplo, si $L = \langle 6, 4, 3, 3, 3, 2, 2, 2\rangle$ y $t = 6$, hay 4 formas distintas de obtener 6 sumando elementos de L:

- 6,
- 4+2,
- 3+3,
- 2+2+2.

L puede contener valores repetidos, por lo que un mismo valor puede ser usado en una suma tantas veces como sea necesario, siempre respetando la cantidad de veces que éste aparece en L (en el ejemplo anterior, vea los casos 2+2+2 y 3+3). Use backtracking recursivo para resolver el problema.

2.1. Formato de Entrada

La entrada de datos será a través de la entrada estándar (`stdin`), y contendrá varios casos de prueba. Cada caso de prueba ocupará una línea separada de la entrada, y tendrá el siguiente formato. Comienza con el valor entero t ($1 \leq t \leq 1000$), seguido por el entero n ($0 \leq n \leq 20$), indicando el tamaño de la lista L. A continuación le siguen los n valores enteros $x_1, \ldots, x_n$ que conforman la lista. Se cumple que $1 \leq x_i \leq 100$. Los valores dentro de una línea de la entrada están separados por

un único espacio. Un valor $n = 0$ indica el fin de la entrada. Los valores $x_1, \ldots, x_n$ están ordenados de forma no creciente, y pueden haber valores repetidos.

Un ejemplo de entrada es el siguiente:

```
6 8 6 4 3 3 3 2 2 2
8 6 8 6 4 4 2 2
10 3 4 2 2
800 12 100 100 100 100 100 100 50 50 50 50 50 50
0 0
```

Hint: para probar su programa de una mejor manera, ingrese los datos de entrada con el formato indicado en un archivo de texto (por ejemplo, el archivo `input-1.dat`). Luego, ejecute su programa desde la terminal, redirigiendo la entrada standard como a continuación, evitando tener que entrar los datos manualmente cada vez que prueba su programa:

```
./problema1 < input-1.dat
```

2.2. Formato de Salida

La salida se hará a través de la salida estándar (`stdout`). Debe mostrar el resultado para cada uno de los casos de prueba dados en la entrada, con el siguiente formato. Por cada caso, debe imprimir una línea que contenga el texto "`Suma de`", seguido por el valor de t correspondiente, seguido por el caracter '`:`'. Luego, se imprimen todas las maneras de sumar t encontradas por el algoritmo. Las sumas deben estar ordenadas de forma no creciente de acuerdo a los números que la conforman. Aún cuando la lista de entrada puede contener elementos repetidos, todas las sumas mostradas deber ser distintas: no se admiten sumas repetidas. Si para algún caso de prueba no hay ninguna manera de sumar elementos de la lista que permita obtener t, se debe imprimir "`NADA`".

La salida correspondiente al ejemplo mostrado anteriormente es:

```
Suma de 6:
6
4+2
3+3
2+2+2
Suma de 8:
8
6+2
4+4
4+2+2
Suma de 10:
NADA
Suma de 800:
100+100+100+100+100+100+50+50+50+50
100+100+100+100+100+50+50+50+50+50+50
```

3. Problema 2: Terrenos Inundados

Una compañía constructora ha comprado una gran área de terreno en el sur del país. Dadas las condiciones meteorológicas y geográficas de la zona, el terreno posee lagunas. Aunque en principio puede parecer una desventaja, el poseer lagunas internas es un beneficio para el paisaje, y por lo tanto el valor de las propiedades que se construyan será más elevado.

Los ingenieros de la compañia dividieron el terreno en una grilla (o matriz) conformada por celdas de tamaño uniforme. Cada celda contiene ya sea agua o tierra firme, pero no ambas. La pregunta que los ingenieros quieren responder es: dada una celda que contiene agua (identificada por su número de fila y columna en la grilla), ¿Cuál es el área de la laguna que contiene a dicha celda? En este caso, el área de una laguna se calcula como la cantidad de celdas que la conforman. Las celdas diagonales son consideradas adyacentes. Su solución debe ser lo más eficiente posible.
Hint: Modele el problema usando grafos, y resuelva usando algún tipo de recorrido sobre el mismo.

3.1. Formato de Entrada

Los datos serán leídos desde la entrada standard, en donde cada línea tendrá el siguiente formato. La primera línea contiene un único entero positivo, indicando el número de casos de prueba que le siguen. Luego de la primera línea hay una línea en blanco. Además, hay una línea en blanco después de cada caso de prueba. Cada caso de prueba consiste de $0 < n \leq 999$ líneas, cada una conteniendo una secuencia de caracteres 'L' (tierra firme) y 'W' (agua), de largo $0 < m \leq 999$ cada una. Estas n líneas representan a la grilla de $n \times m$. Luego le siguen $k > 0$ líneas, cada una de las cuales contiene un par de enteros i y j, que representan a una celda de la grilla que contiene agua, y serán usados para hacer las consultas sobre la matriz. Un ejemplo particular de entrada es el siguiente:

```
2

LLWLLLLLLLLLLLLWLLL
WLLLLLLLWLLLLWWLWLLL
LLWWLLLLWWLWLWLLLLWL
LWWLLLWWLWLLLLLWLLLL
4 2
2 1
1 3
2 9

WLL
LLL
LWL
WLW
LWW
LLL
1 1
3 2
```

3.2. Formato de Salida

La salida del programa debe mostrarse a través de la salida standard. Para cada caso de prueba, la salida debe seguir la siguiente descripción. Las salidas de dos casos de prueba consecutivos deben ser separados por una línea en blanco.

Para cada par de enteros i y j presentados como entrada, se debe imprimir una línea que indique el área de la laguna que contiene a la celda con fila i y columna j de la grilla.

```
4
1
1
6

1
5
```

4. Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea3-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda), o alternativamente usando formato zip, en el sitio Aula USM del curso, a más tardar el día 12 de noviembre, 2020, a las 23:59:00 hrs (Chile Continental), el cual contenga:

- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, Nombre y ROL de cada integrante del grupo.
- `README.txt`, Instrucciones de compilación en caso de ser necesarias.
- `Makefile`, Instrucciones para compilación automática.