Laboratorio de Programación - Labo11

Taller de Ordenamiento

Algoritmos y Estructuras de Datos I

Departamento de Computación, FCEyN, Universidad de Buenos Aires.

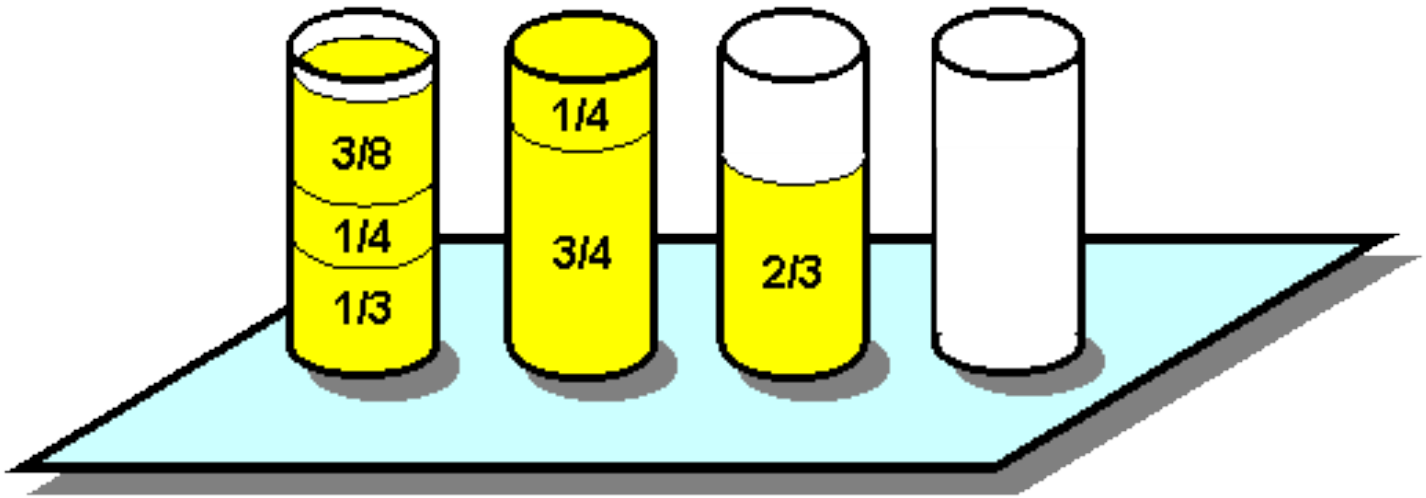
Menú del día

- Programación, Programación, Programación:
  - BPP (Bin Packing Problem)
  - Anagramas

Primer Problema

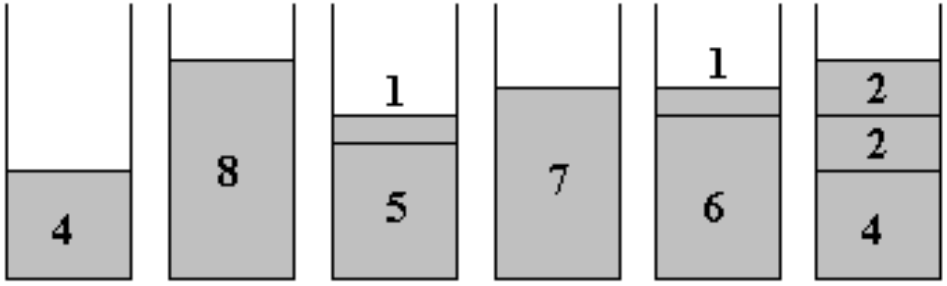
Bin Packing Problem

Dada una lista de objetos junto a sus volúmenes y una lista de contenedores de dimensión fija, encontrar el mínimo número de contenedores para que todos los objetos sean asignados a algún contenedor.



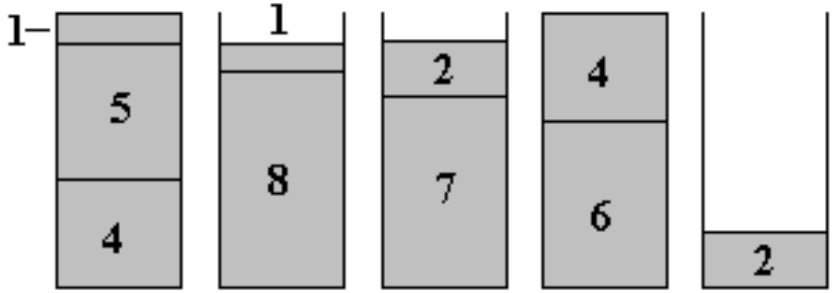
BPP

- Es un problema muy estudiado, de la clase NP-Hard (algo3)
- Encontrar la solución exacta es computacionalmente difícil
- Podemos intentar encontrar soluciones aproximadas utilizando heurísticas
  - Next Fit: Ponemos los números de acuerdo al orden en el que vienen.
  - Para un conjunto S = {4, 8, 5, 1, 7, 6, 1, 4, 2, 2} sobre contenedores de volumen 10



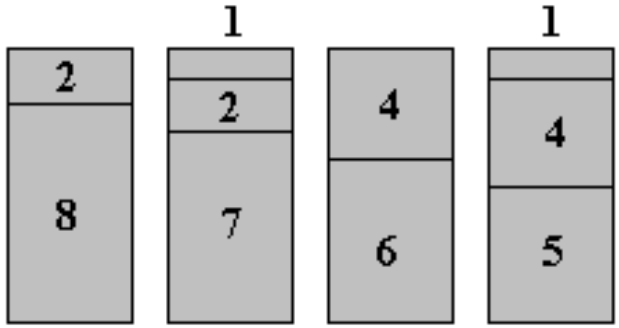
BPP

- Es un problema muy estudiado, de la clase NP-Hard (algo3)
- Encontrar la solución exacta es computacionalmente difícil
- Podemos intentar encontrar soluciones aproximadas utilizando heurísticas
  - Best Fit: Ponemos los números que mejor se ajustan al volumen disponible, según el orden en que vienen.
  - Para un conjunto S = {4, 8, 5, 1, 7, 6, 1, 4, 2, 2} sobre contenedores de volumen 10



BPP

- Es un problema muy estudiado, de la clase NP-Hard (algo3)
- Encontrar la solución exacta es computacionalmente difícil
- Podemos intentar encontrar soluciones aproximadas utilizando heurísticas
  - Best Fit Ordenado: Ordenamos y ponemos los números que mejor se ajustan al volumen disponible, según el orden en que vienen.
  - Para un conjunto S = {8, 7, 6, 5, 4, 4, 2, 2, 1, 1}



BPP - Programando

- Disponemos de un código, el cual debemos completar (BPP.cpp)
- El programa toma como parámetro un archivo de texto
  - El formato es #números, volumen del contenedor, seguido de una lista de valores (para este labo, los contenedores tienen una capacidad de 150).
- El programa se encarga de ejecutar Best Fit antes y después de ordenar la lista.
- Contamos con 4 archivos de volúmenes para ordenar, desde 1.000 elementos a 1.000.000.
- Calcular el tiempo de cómputo empleado por las implementaciones de ordenamiento para cada valor de entrada. Proponer un orden de complejidad del método. (Usar las funciones vistas en el laboratorio de búsqueda).

¡Un momento! ¿Cómo ordeno?

Tenemos varias formas de hacerlo:

- Insertion Sort
- Selection Sort
- Bubble Sort
- Merge Sort
- Heap Sort
- Quick sort
- Bogo Sort
- Timsort
- etc. sort

Repasando lo visto: Insertion y Selection

Idea

- Insertion Sort: *Inserto* ordenado al principio del vector, buscando la posición que le corresponde.
- Selection Sort: *Selecciono* el minimo del vector y lo posiciono donde correspone.

Segundo Problema

Anagramas

El DC necesita de nuestra ayuda. De tanta gente que vino a la Semana de la Computación se perdió la implementación del stand Anagramas y el departamento sabe que en algoritmos 1 está el futuro de la computación, por eso nos pidieron ayuda.

- Una palabra es anagrama de otra si las dos tienen las mismas letras, con el mismo número de apariciones.
- Por ejemplo la palabra "delira" es anagrama de la palabra "lidera".

Anagramas

- Se pide implementar un programa que tome una palabra ingresada por el usuario e imprima por pantalla todas las palabras del idioma que son anagrama de esa.
- Como esta implementación puede tener repercusiones educativas internacionales vamos a hacerlo para el idioma ingles, usando palabras que solo tienen letras minusculas de la "a" a la "z".

Dos implementaciones distintas

Usando ordenamiento

Para ver si p1 y p2 son anagramas basta con ordenar los caracteres de cada una y comparar si el string resultante es el mismo.

Dos implementaciones distintas

Usando números primos

- Asignamos a cada letra del alfabeto un número primo distinto.
- Calculamos los números primeros correspondientes a las letras de la palabra
- Multiplicamos estos números y obtenemos un número que identifica a la palabra.
- Por el teorema fundamental del álgebra a cada combinación de letras se le asigna un número que identifica unívocamente a todos los anagramas.
- Entonces alcanza con comparar si los números obtenidos son iguales.

A programar

- Crear un archivo anagramas.cpp y programar cada una de las implementaciones de esAnagrama(). Agregarlo al CMakeList.txt.
- Para probar: el main toma una palabra en ingles y chequea la cantidad de anagramas que hay en el diccionario.