Estructuras de datos 2024-2 (examen 1)

Universidad tecnológica de Pereira

Profesor: Gustavo Gutiérrez

Pregunta 1

La clase `vector` implementa lo que se conoce como un *arreglo dinámico*. Es decir, en parte se comporta como un arreglo pero *extiende* la funcionalidad de los arreglos permitiendo que la estructura de datos crezca o se contraiga de acuerdo a su uso. Responda las siguientes preguntas respecto a la estructura de datos *vector*.

1. Por qué es necesario hacer resize del vector cuando este se llena y es necesario seguir almacenando elementos?. A caso no sería una mejor opción apartar un espacio en memoria para seguir almacenando los elementos que se van a insertar en el futuro así se deba llevar registro dentro de la clase de que los elementos del vector pueden estar almacenados en diferentes lugares de la memoria?
2. Si comparamos el vector con una lista enlazada como la implementada en el curso cree usted que la lista pueda ser más eficiente?. Justifique su respuesta.

Pregunta 2

Considere el siguiente programa:

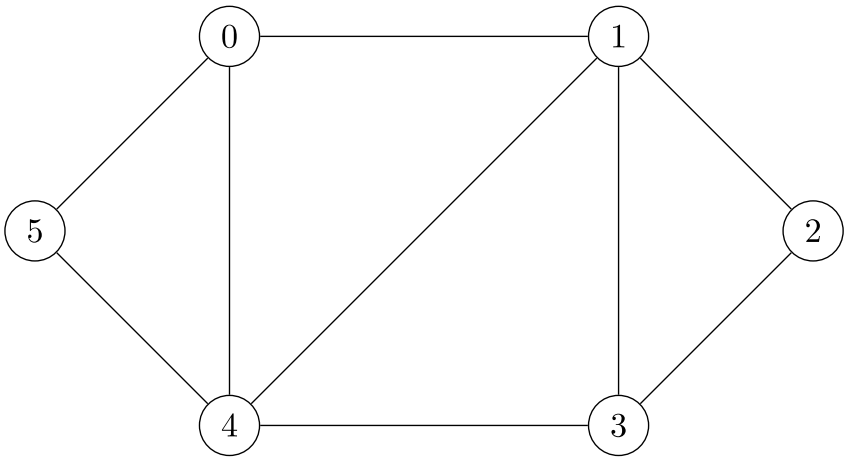
```
int main() {
        Vector<int> x;
        Vector<int> y(10,1.8);
        Vector<int> z(100, 2.0);
        for (unsigned int i = 0; i < 10000; i++) {
                x.push_back(i);
                y.push_back(i);
                z.push_back(i);
        }
        return 0;
}
```

El vector `x` es construido (línea 2) utilizando el constructor por defecto que tiene una capacidad inicial de 5 y una política de crecimiento de 2.0. Es decir, cada que se redimensiona se solicita memoria para un arreglo del doble de los elementos que se almacenan hasta ese momento. El vector `y` se construye (línea 3) con 10 elementos de capacidad inicial y una política de crecimiento de 1.8. Por último el vector `z` es inicializado (línea 4) con una capacidad inicial de 100 elementos y una política de crecimiento de 2.0.

1. Para cada uno de los vectores especifique cuál es su capacidad final y el desperdicio en el que incurre.
2. Cuál de los vectores resultó ser más eficiente a la hora de ejecutar el programa? Justifique clara y concisamente su respuesta.

Pregunta 3

Se cuenta con la siguiente representación de una red de transporte. Los círculos representan ciudades y las líneas representan conexiones entre ellas. Pro ejemplo, la ciudad representada por 0 se encuentra conectada con las ciudades 1, 4 y 5.



En este punto usted deberá crear la estructura de datos `AdjacencyList` para representar redes de transporte de forma general. Es decir, con un número arbitrario de ciudades y con conexiones arbitrarias entre las mismas. Tenga en cuenta que las ciudades siempre se representan con números consecutivos que comienzan desde cero. Además es importante notar que no todas las ciudades están conectadas entre si.

1. Proponga una implementación en C++ de la clase `AdjacencyList`, defina apropiadamente sus atributos y sus operaciones. Usted puede utilizar para esto cualquiera de las estructuras de datos discutidas en clase.
2. Cree operaciones para adicionar y remover ciudades.

3. Cree una operación que dada una ciudad retorne las ciudades que se encuentran conectadas con esta.