Práctica 3: Algoritmos Voraces

Algorítmica

Pablo Mariano Moreno Mancebo

(76067676)



Descripción del trabajo - Ejercicio 3

Se han organizado cursos de esquí en Sierra Nevada para los que se dispone de esquíes de diversos tamaños. A cada alumno se le deben asignar unos esquís cuyo tamaño sea lo más cercano posible a su altura. Sabiendo las alturas de los alumnos y los tamaños de los esquís disponibles, es necesario hacer una asignación de los mismos tal que el error cuadrático medio sea mínimo. Suponiendo que hay N alumnos y M esquís de longitud L(j) (1≤j≤M), H(i) es la altura del alumno i, A(i) es la longitud del esquí asignado al alumno i, el error cuadrático medio se calcula de la siguiente forma:

Índice

[Análisis](#_3kt1szcb8ib1)

[Búsqueda del problema a resolver y descripción del problema](#_8v1otmx2deqd)

[Diseño](#_rjclt4286hrb)

[Componentes Greedy](#_aanqzrea84ys)

[Adaptación de plantilla al algoritmo](#_3g58zavkfemu)

[Implementación](#_xz8k54v6rw4j)

[Anexo : Archivo data.txt](#_slsoxcw40c0v)

# Análisis

## Búsqueda del problema a resolver y descripción del problema

El problema de los esquís para un grupo de alumnos : Tenemos que contar con las alturas de todos los alumnos , debemos disponer de un número de esquís mayor o igual al número total de alumnos. Para resolver el problema y llevar el control simplemente le asignamos a cada alumno un esquí superior o igual a su talla , y vamos acumulando el error para controlarlo ( talla dada - altura del alumno).

Como se puede controlar el error según el planteamiento anterior podemos aplicar un algoritmo greedy donde en podemos asociar en cada una iteración la mejor opción al alumno dentro de las posibilidades.

# Diseño

## Componentes Greedy

* lista de candidatos: Los esquís posibles a devolver
* lista de candidatos ya utilizados: Los esquís que se han ido seleccionado para ser devueltos.
* función solución: La suma de los errores en la asociación de esquís-alumno es el error cuadrático medio que se devuelve.
* criterio de factibilidad: Los esquís disponibles son mayor que la altura del alumno
* función de selección: El esquí de menor valor que se le resta la altura del alumno seleccionado
* función objetivo: Minimizar el error cuadrático medio para el problema

## Adaptación de plantilla al algoritmo

| **Función S= Voraz(vector candidatos C)**  asignacion = Ø ; error = 0 ; pos\_alum = pos \_ talla = -1  **Mientras** (!vec\_alumnos.empty()) **hacer:**  pos\_alum=0  **Si** existe esqui > altura\_alumno[pos\_alum] **entonces**  pos\_talla != -1  **en otro caso**  **Devolver** “No hay solución”  **Fin-Si**  pos\_talla = el menor elemento de esquís tal que esquis[pos\_talla] < alumno\_altura - vec\_esquis  error = error + (vec\_esquis[pos\_talla] - vec\_alumnos[pos\_alumno])²  asignacion = asignacion U {pos\_alum, pos\_talla}  vec\_esquis = vec\_esquis \{pos\_talla}  vec\_alumnos = vec\_alumnos \{pos\_alum}  **Fin-Mientras**  **Devolver** error/num\_alumnos |
| --- |

## 

# Implementación

**compilar :**

g++ -O2 -o ejecutable codigo.cpp

**ejecución:**

./ejecutable data.txt

//Adjunto el archivo data.txt para poder modificar los datos y sea más fácil la lectura de datos

| vector<pair<string,int>> greedy(vector<Alumno> alumnos,vector<int> esquis)  {  vector<pair<string,int>>asignacion;  double error\_cuadratico =0;  int num\_alumnos=alumnos.size();    //Borramos el vector  asignacion.clear();  int it=0;  while (it<num\_alumnos)  {  int pos\_alumno,pos\_talla=-1;  //Comprobamos si hay solución posible con los esquis actuales  int ini=-1;  int k=0;  while (k<=esquis.size() && ini==-1)  {  if (esquis[k]>=alumnos[0].altura){  ini=k;  }  k++;  }  //Si no encontramos esquis disponibles se termina el programa  if (ini ==-1)  {  cout<<"No se ha encontrado solución."<<endl;  asignacion.clear();  return asignacion;  }  double minimo = alumnos[0].altura - esquis[ini];  //Busco el esqui con el cual la altura - la talla al cuadrado sea minimo entre los esquis no descartados  double aux;  pos\_talla=ini;  for (int i = ini+1; i < esquis.size(); i++)  {  if (alumnos[0].altura- esquis[i] >= 0) // Comparo con una talla superior a la altura  {  aux = alumnos[0].altura - esquis[i];  if (aux < minimo && minimo != aux) // Comparo con el que minimo actual y actualizo variable  {  minimo=aux;  pos\_talla=i;  }  }  }  //Guardo la id del alumno junto a la talla que se le ha asignado  asignacion.push\_back(make\_pair(alumnos[0].id,esquis[pos\_talla]));  //Elimino al alumno y la talla de sus respectivos vectores  eliminar\_pos\_vec(0,alumnos);  eliminar\_pos\_vec(pos\_talla,esquis);  //Acumulo el error  error\_cuadratico+= pow(minimo,2);  //Incremento el numero de alumnos con esquis asignados  it++;  }  //Calculo del error cuadratico medio  string msg= "error\_medio= " + to\_string(error\_cuadratico/num\_alumnos);  cout <<msg<<endl;  return asignacion;  } |
| --- |

| 1. **Pruebas : Ejemplo de ejecución del algoritmo** |
| --- |

**compilar :**

g++ -O2 -o ejecutable codigo.cpp

**ejecución:**

./ejecutable data\_prueba.txt

//Adjunto el archivo data\_prueba.txt para obtener los mismos datos

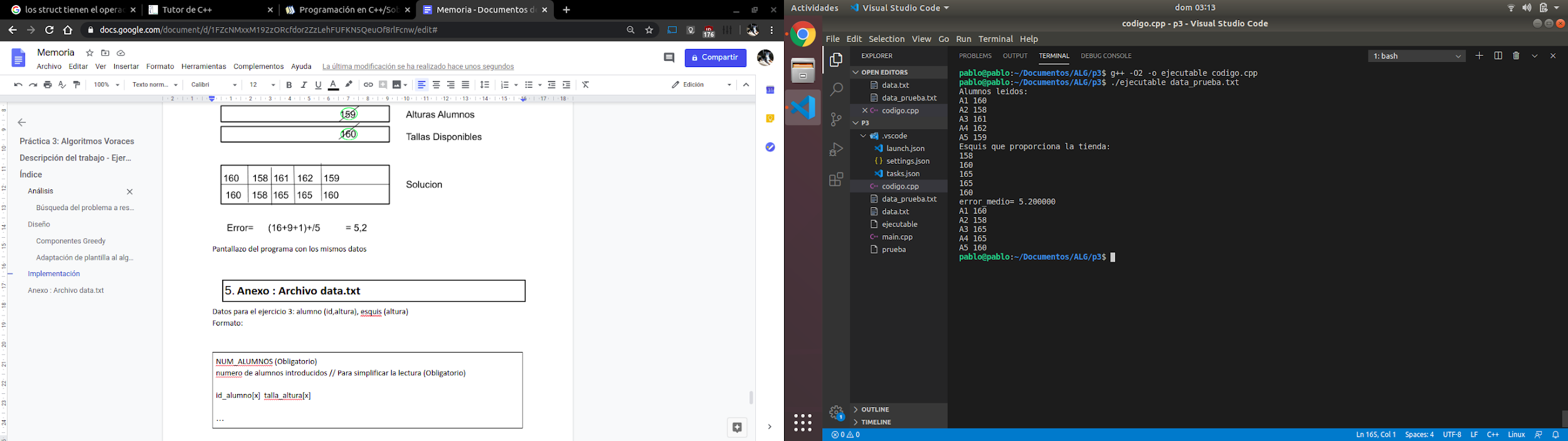








Pantallazo del programa con los mismos datos



# Anexo : Archivo data.txt

Datos para el ejercicio 3: alumno (id,altura), esquis (altura)

Formato:

| NUM\_ALUMNOS (Obligatorio)  numero de alumnos introducidos // Para simplificar la lectura (Obligatorio)  id\_alumno[x] talla\_altura[x]  …  NUM\_ESQUIS(Obligatorio)  numero de esquis disponibles de la tienda // Para simplificar la lectura (Obligatorio)  talla[x]  …. |
| --- |