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: $E = \frac{1}{N} \sum_{i=1}^{N} (H(i) - L(i))^2$

Índice

<u>Análisis</u>

Búsqueda del problema a resolver y descripción del problema

<u>Diseño</u>

<u>Componentes Greedy</u>
<u>Adaptación de plantilla al algoritmo</u>

<u>Implementación</u>

Anexo: Archivo data.txt

1. 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.

2. 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])<sup>2</sup>
 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
```

3. 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>
  vector<pair<string,int>>asignacion;
  int num alumnos=alumnos.size();
  asignacion.clear();
       int pos_alumno,pos_talla=-1;
       while (k<=esquis.size() && ini==-1)</pre>
           if (esquis[k]>=alumnos[0].altura) {
```

```
cout<<"No se ha encontrado solución."<<endl;</pre>
           asignacion.clear();
       double minimo = alumnos[0].altura - esquis[ini];
       pos talla=ini;
       for (int i = ini+1; i < esquis.size(); i++)</pre>
           if (alumnos[0].altura- esquis[i] >= 0) // Comparo con una
               aux = alumnos[0].altura - esquis[i];
                   minimo=aux;
                   pos talla=i;
asignacion.push back(make pair(alumnos[0].id,esquis[pos talla]));
       eliminar pos vec(0,alumnos);
       eliminar_pos_vec(pos_talla,esquis);
       error cuadratico+= pow(minimo,2);
```

```
//Calculo del error cuadratico medio
string msg= "error_medio= " +
to_string(error_cuadratico/num_alumnos);
cout <<msg<<endl;
return asignacion;
}</pre>
```

4. Pruebas : Ejemplo de ejecución del algoritmo

compilar:

g++ -O2 -o ejecutable codigo.cpp

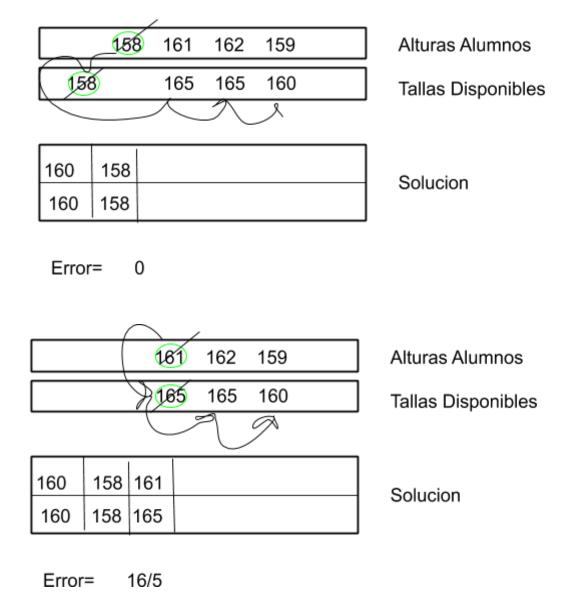
ejecución:

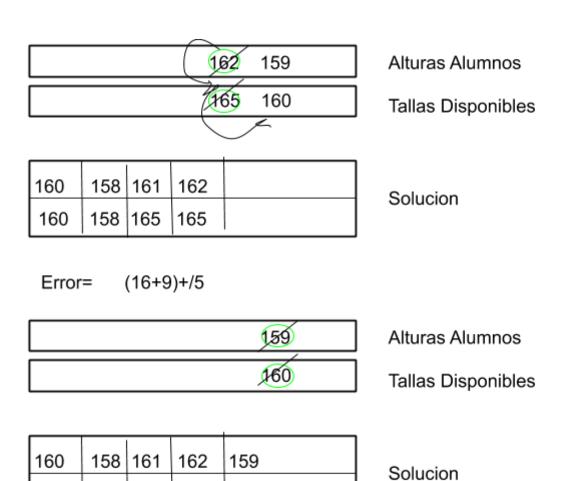
./ejecutable data_prueba.txt

//Adjunto el archivo data_prueba.txt para obtener los mismos datos

160 158 161 162 159	Alturas Alumnos
158 160 165 165 160	Tallas Disponibles
	Solucion
Error= 0 160 158 161 162 159 158 160 165 165 160	Alturas Alumnos Tallas Disponibles
160	Solucion

Error= 0





Error= (16+9+1)+/5 = 5,2

165

160

Pantallazo del programa con los mismos datos

165

158

160

```
PROBLEMS OUTPUT
                        TERMINAL
                                     DEBUG CONSOLE
pablo@pablo:~/Documentos/ALG/p3$ g++ -02 -o ejecutable codigo.cpp
pablo@pablo:~/Documentos/ALG/p3$ ./ejecutable data_prueba.txt
Alumnos leidos:
Al 160
A2 158
A3 161
A4 162
A5 159
Esquis que proporciona la tienda:
158
160
165
165
160
error medio= 5.200000
Al 160
A2 158
A3 165
A4 165
    160
pablo@pablo:-/Documentos/ALG/p3$
```

5. 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]
····