

# MC458 Projeto e Análise de Algoritmos I

Primeiro semestre de 2016

Docente: Orlando Lee

Monitora PED: Amanda Resende

## O problema da moradia

Phil, após os eventos fantásticos ocorridos no filme *Groundhog Day* (Feitiço do Tempo em português), decidiu que iria morar junto com Rita em Punxsutawney, Pennsylvania. **Observação:** você não precisa ter assistido o filme para resolver este laboratório.

Como ele se tornou amigo de várias pessoas da cidade, ele decidiu que iria morar em algum lugar que fosse tão “próximo de todos” quanto possível. Como a cidade era relativamente pequena, todos seus amigos viviam na avenida principal da cidade (só havia uma). Suponha que os amigos de Phil são identificados por  $1, 2, \dots, N$ . Imagine que a avenida seja a reta real. Cada amigo  $i$  tem um endereço  $x_i$  que corresponde a um ponto na reta. Pense que cada  $x_i$  corresponde a um quarteirão da avenida. Assim, dois amigos podem ter o mesmo endereço.

Phil, apesar de tratar todos seus amigos igualmente bem, tinha uma preferência maior por alguns amigos do que outros (ninguém é perfeito, muito menos Phil). Para cada amigo  $i$ , Phil atribui um **peso real positivo**  $w_i$  a este, representando o quão próximo ele quer ficar próximo deste. A soma total dos pesos é  $wt = \sum_{i=1}^N w_i$ .

Phil gostaria de escolher um endereço  $x$  tal que a soma  $\sum_{i=1}^N w_i \cdot d(x, x_i)$  fosse a menor possível, onde  $d(x, x_i)$  é a distância de  $x$  a  $x_i$ , ou seja,  $|x - x_i|$  (valor absoluto de  $x - x_i$ ).

Um problema correlato ao problema do Phil é o **problema da mediana ponderada**. Neste problema são dados  $N$  elementos  $x_i$  cada com um **peso real (no nosso caso, entre 0 e 1000)**  $w_i$  **associado**. Deseja-se encontrar um elemento  $x_m$  tal que:

$$\sum_{i: x_i < x_m} w_i < \frac{wt}{2}$$

e

$$\sum_{i: x_i > x_m} w_i \leq \frac{wt}{2}.$$

Veja o problema 9-2 no CLRS. **Note que os elementos não precisam estar ordenados.**

Por exemplo, se  $N = 3$ ,  $x_1 = 0$ ,  $x_2 = 5$ ,  $x_3 = 1$ ,  $w_1 = 0.2$ ,  $w_2 = 0.6$ ,  $w_3 = 0.2$  então a melhor solução é  $x = 5$ . Parece estranho que a solução seja exatamente o endereço de um amigo, mas isto significa apenas que Phil irá morar no mesmo quarteirão. Se os pesos fossem  $w_1 = 0.3$ ,  $w_2 = 0.4$ ,  $w_3 = 0.3$  então a melhor solução seria  $x = 1$ . Uma instância pode ter várias soluções ótimas, mas sempre tem uma que corresponde a escolher um dos pontos originais.

## Entrada

A entrada consiste de 3 linhas. A primeira contém um inteiro  $N$  tal que  $1 \leq N \leq 100.000$  representa o número de amigos de Phil. A segunda linha possui  $N$  inteiros não-negativos

distintos que representam os endereços dos amigos de Phil. Por fim, a terceira linha contém  $N$  reais positivos tal que  $0 \leq N \leq 1000$  que representam a ponderação feita por Phil.

## Saída

Um ponto  $p$  que minimiza a soma  $\sum_{i=1}^N w_i \cdot d(x, x_i)$ .

### Exemplo 1

```
3                /* entrada */
0 5 1           /* entrada */
0.2 0.6 0.2     /* entrada */
5               /* saída */
```

### Exemplo 2

```
4
10 20 50 10
0.1 0.2 0.6 0.1
50
```

### Exemplo 3

```
10
1 4 8 100 7 20 500 4 20 10
0.05 0.05 0.05 0.3 0.1 0.05 0.1 0.05 0.05 0.2
10
```

### Exemplo 4

```
10
2 6 8 3 1 1 3 5 3 4
3 9 10 7 4 6 5 2 1 8
4
```

### Exemplo 5

```
10
2 6 8 3 1 1 3 5 7 4
3 9 10 7 4 6 5 2 7 2
5
```

## Especificações

1. O programa deve ser implementado em C ou C++. A versão dos processadores usados pelo SuSy é **C99** e **C++** respectivamente. **Seu programa deve ser compilável nestas versões.**
2. Não serão aceitas soluções que ordenam o “vetor”  $x$ . Você pode usar qualquer um dos outros algoritmos que resolve o problema da seleção vistos em aula.

## Avaliação

Haverá 15 testes abertos e 5 teste fechados. A nota do projeto é proporcional ao número de teste bem-sucedidos. Mais precisamente, se  $N$  é o número de testes bem-sucedidos do seu programa, então sua nota será  $(\frac{N}{20}) \times 10 = \frac{N}{2}$ .

## Prazo de submissão

O programa pode ser submetido até as seguintes datas:

- até 23:59h de 08 de maio (domingo) - sem penalidade,
- até 23:59h de 09 de maio (segunda) - serão descontados  $-4$  pontos da nota, ou seja, a nota será  $(\frac{N}{2}) - 4$ .

## Observações

1. Seu algoritmo deve ser eficiente. Haverá um limite de tempo máximo (1 segundo) para as instâncias de tamanho 100.000.
2. O número máximo de submissões é 10.
3. Para a avaliação será considerada apenas a última versão do programa submetido.
4. Para submissão no Susy utilize o número do seu RA em **Usuário** e a senha utilizada na DAC em **Senha**.
5. Note que os algoritmos vistos em sala para o problema da seleção supõe que o vetor original é indexado de 1 a  $N$ . Tome algum cuidado na sua implementação.
6. É provável que você encontre na Web uma descrição de como resolver o problema, se você não se deu ao trabalho de pensar. Você pode consultar essas fontes. Entretanto você não pode aproveitar código da Web ou do seu colega. Isto será considerado **plágio**. A única exceção é que você pode aproveitar código que gere números aleatórios, caso você queira usar Select Aleatorizado. Mencione no arquivo .c/.cpp de qual site você tirou o código.