Trabalho Prático 2 — Ordenação de Sistemas Solares

1 Introdução

Na terra da Computação, os avanços tecnológicos estão a todo vapor. Os cientistas desenvolveram um novo dispositivo, capaz de identificar estrelas: o telescoptero. Como se trata de um dispositivo muito simples e divertido de usar, ele passou a ser amplamente adotado pelos computistas. O problema? Com a constante descoberta de novos sistemas solares, os cientistas não sabem por quais astros eles devem começar seus estudos.

No entanto, existe um consenso entre os astrônomos das características que tornam um sistema interessante. Sua tarefa é, ordenar os sistemas solares recém descobertos com base nessas características, assim salvando a vida de milhares de acadêmicos. O governo da Computolândia pretende cortar a verba de pesquisa caso os cientistas não façam uma grande descoberta em breve. Ou seja, será necessário que **você implemente um algoritmo rápido** para a ordenação.

2 Informações Adicionais

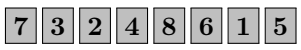
O consenso entre os cientistas é que um **sistema solar A é MAIS interessante que um sistema solar B, se:**

1. O raio do sol do sistema A é maior que o raio do sol do sistema B.
2. O sistema A possui mais planetas que o sistema B.
3. O maior planeta do sistema A é maior que o maior planeta do sistema B.
4. O sistema A possui mais luas que o sistema B.
5. A maior lua do sistema A é maior que a maior lua do sistema B.
6. O tempo de descoberta de A é **menor** que o de B.

Perceba que os critérios estão ranqueados, ou seja: na comparação entre dois sistemas, os critérios são comparados em ordem. Por exemplo, se um sistema x possui 100 planetas, mas o raio de seu sol é 5, ele é menos interessante que um sistema y que possui 1 planeta, mas cujo sol tem raio 999, pois o raio é o primeiro critério. Dessa maneira, **os critérios subsequentes SÓ SÃO usados** em caso de empate dos critérios anteriores. Assim, o critério de comparação da maior lua somente será usado se ambos os sistemas possuírem o mesmo raio dos sóis, a mesma quantidade de planetas, o mesmo tamanho máximo de planetas e também a mesma quantidade de luas. É possível que mesmo com todos esses critérios ainda possa haver empate; nesse caso sua função de comparação deve levar em conta o sexto critério: o tempo de descoberta. Como a Computolândia possui apenas um telescoptero, o tempo de descoberta de cada sistema é único e não existem empates nesse caso.

E falando em algoritmo, para atender à demanda dos cientistas de que a classificação dos sistemas deve ser rápida, você deverá implementar o algoritmo desenvolvido pelo astrônomo e cientista da computação Bromero Rito: o *bromerosort*. Ele funciona da seguinte maneira: divida o vetor em 2 subvetores de aproximadamente o mesmo tamanho sucessivamente, até chegar no caso base: um vetor com apenas 2 elementos. Como comparar 2 elementos é trivial, ordene esses elementos. Em seguida, "suba", combinando os pares em quartetos ordenados, os quartetos em octetos e assim por diante. De fato, a magia do *bromerosort* está no fato de que essa combinação pode ser feita de forma super eficiente.

Para ilustrar o funcionamento do algoritmo, vamos ordenar uma lista com 8 inteiros:



Dividindo o vetor em 2:

Como o tamanho dos vetores ainda não é 2, é necessário dividir novamente

Finalmente o tamanho dos vetores é 2. Podemos agora fazer a comparação trivial e ordená-los:

Perceba que os pares que já estavam ordenados não foram alterados. Agora é a hora de fazer a combinação, que o herói Bromero Rito chamou de *ritação*. Vamos começar com os dois vetores da esquerda. Seja i o índice do primeiro vetor e j o índice do segundo vetor, ambos iniciados com zero. Fazemos a comparação dos índices correspondentes: 3 é menor 2? Não! Então vamos inicializar nosso vetor combinado com 2 e vamos incrementar o índice **somente do segundo vetor** ($j = 1$). Comparando então 3 e 4, 3 realmente é menor. Portanto, 3 entra no vetor combinado e incrementamos o índice apenas do primeiro vetor.

Comparando 7 e 4, 4 é menor, então ele entra na combinação. Como o primeiro vetor acabou, podemos simplesmente incluir tudo que restava nele no final da combinação. Podemos ilustrar esse processo da seguinte forma.

Passo a passo da *ritação*

Índices

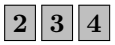
Combinação

Índices

Combinação

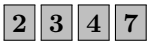
Índices

Combinação



Índices

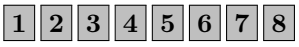
Combinação



Versão final:

Replicando o mesmo processo para os dois vetores da direita, ficamos com:

Por fim, bastar ritar os dois vetores de tamanho 4:



Uma dúvida natural seria: "nesse caso o vetor tinha um tamanho que era uma potência de base 2, então tudo funcionou perfeitamente. Mas e se o tamanho fosse, por exemplo, 9?" Não tema, o grande Bromero Rito pensou nisso. Se o vetor de exemplo fosse:

Bastaria dividir o vetor em 2:

E seguir como no caso anterior. A realidade é que existe um caso ainda mais fácil que comparar 2 elementos: ter apenas um elemento. Neste novo exemplo, a divisão do vetor de 8 elementos ocorreria normalmente, depois ele seria ordenado usando as combinações e por fim, combinado de novo com o elemento 5.

Um jeito de visualizar esse processo é através do seguinte pseudocódigo:

```
1  BromeroSort(sistemas, esquerda, direita) {
2    tamanho = direita - esquerda
3    se tamanho == 1 {
4      retorne // sistema ja ordenado.
5    }
6    se tamanho % 2 == 1 {
7      BromeroSort(sistemas, esquerda, direita - 1)
8      Ritacao(sistemas, esquerda, direita - 1, direita -1, direta)
9      retorne
10   }
11   meio = (direita + esquerda) / 2
12   BromeroSort(sistemas, esquerda, meio)
13   BromeroSort(sistemas, meio, direita)
14   Ritacao(sistemas, esquerda, meio, meio, direita)
15 }
```

Nele, o processo de *ritação* altera diretamente o conteúdo do vetor sistemas, não sendo necessário retornar nada dessa subrotina. As variáveis `esquerda` e `direita` dizem para o *BromeroSort* que devemos ordenar o vetor `sistemas` entre os índices `esquerda` e `direita - 1`. Ou seja, a chamada *BromeroSort(sistemas, 2, 6)* deve ordernar o subvetor do vetor `sistemas` composto pelos elementos nos índices 2,3,4, e 5. O processo de *ritação* é bem simples, e Bromero Rito tem plena confiança que qualquer pessoa que leu sobre seu glorioso método é capaz de implementar a ritação por conta própria.

3 Entrada

Convenientemente, os cientistas te enviaram a lista de sistemas solares em um formato padronizado. Primeiro você deve ler a quantidade de sistemas solares, N. Depois você deve ler o tempo de descoberta do sistema T_i, nome do sistema S_i, o raio do seu sol R_i, e a quantidade de planetas que ele possui P_i, tudo em uma linha. Em seguida, é a hora de ler os planetas *daquele* sistema: o nome do planeta, $p_{i,j}$, o raio do planeta $r_{i,j}$ e a quantidade de luas, $L_{i,j}$. Por fim, você deve ler as luas *daquele* planeta: o nome da lua $\ell_{i,j,k}$ e seu raio $q_{i,j,k}$. Perceba que assim que você terminar de ler as luas de um planeta, você vai ler o próximo planeta e suas luas. O mesmo vale para os sistemas: primeiro você lê tudo que tem dentro de um sistema (seus planetas e luas) e **depois** lê o próximo sistema.

A entrada possui as seguintes restrições. Para facilitar, os nomes, raios e quantidades possuem as mesmas restrições[1], ou seja, não existe uma restrição quanto ao raio dos planetas ou das luas, e sim uma restrição de *raios*, no geral. A exceção é o número de sistemas, S, que pode ser maior que o número de planetas / luas (indicados por *length*).

- $1 \leq N \leq 1000000$.
- $1 \leq T_i \leq 1000000$ para todo i.
- todos os nomes tem pelo menos 1 e no máximo 100 caractere s.
- todos os raios são no máximo de tamanho 10000
- cada sistema tem no máximo 100 planetas e cada planeta tem no máximo 100 luas.

A entrada usa *tabs* (`'\t'`) para facilitar a distinção do que é estrela, planeta, ou lua, como se fosse um código em C. Estrelas não possuem nenhum *tab* na frente, planetas apenas 1 e luas, duas tabulações. Você deve considerar a leitura dos `'\t'` quando for fazer seu programa.

[1] Apesar disso não fazer sentido astronômico (bem como o resto do trabalho).

3.1 Exemplo

Não se preocupe com o nome dos astros, eles foram gerados aleatoriamente!

```
6                          // # Sistemas Solares
3 X1dhcdfx35 889 0           // Sistema #1 | Raio | # Planetas
1 nmok0lee2 298 2            // Sistema #2 | Raio | # Planetas
  4V8VP5P 743 2            // Planeta #1 | Raio | # Luas
    SKZJqLg 538            // Lua #1 | Raio
    23VFgWcFM3 978         // Lua #2 | Raio
  ueJSO 805 1              // Planeta #2 | Raio | # Luas
    phuVmf 484             // Lua #1 | Raio
2 jW2Ol 557 1                // Sistema #3 | Raio | # Planetas
  LUTmBvC8C 901 1          // ...
    zzMpnv 636
20 wP30Hu 944 3
  nBSykm3G 187 1
    Hbf1J58 932
  1pUA1xD1w0 980 4
    bfFx 357
    0MkEM 438
    cSLdX 222
    XHbj8C 137
  60yzYmj 316 3
    WZepWXuYaW 471
    u4Fz2 418
    kqYsyi 9
11 kNahmOcQLr 298 3
  71wIO 468 3
    Ryl4D3A 358
    QBPGgLD 659
    6bsIGvPap 311
  8ke2ice 348 0
  YxjJRC 645 4
    r6TwFh 873
    u8bBOSu 736
    3dxmIJWNT 377
    ih6PPWGt 611
40 9YRGgCqY 557 1
  t7ipn3 207 1
    8CwDd 268
```

Saída Esperada

```
wP30Hu      // Maior tamanho
X1dhcdfx35  // Segundo maior tamanho
jW2Ol       // Empate no tamanho e # de planetas, mas planeta MAIOR
9YRGgCqY    // Empate no tamanho e # de planetas, mas planeta menor
kNahmOcQLr  // Empate no tamanho, mas mais planetas
nmok0lee2   // Empate no tamanho, mas menos planetas
```

Obviamente você não deve imprimir os comentários, no exemplo eles são apenas didáticos. Perceba, também que os sistemas devem ser impressos na ordem inversa da ordenação ("menor para o maior"), uma vez que os cientistas estão interessados em estudar os sistemas MAIS importantes ("maiores") primeiro. O nome dos planetas e das luas não deve ser impresso, **apenas** os nomes dos sistemas.