UFMG/ICEx/DCC — Algoritmos e Estrutura de Dados 3

Trabalho Prático 3

Bacharelado — 1º Semestre de 2011

Esse trabalho prático tem por objetivo familiarizar o aluno com paradigmas de projetos de algoritmos, especialmente programação dinâmica e algoritmos gulosos.

1 Definição do problema

Você, um guitarrista famoso, vai fazer um show em Belo Horizonte. Uma das suas estratégias para fazer um show empolgante é nunca tocar duas músicas seguidas no mesmo volume. Cada vez que você troca de música, você faz alguma alteração no volume.

Para o show em BH, você já escolheu quantas e quais músicas vai tocar, e também o quanto o volume deve variar (subir ou descer) de uma música para a outra. Seu problema agora é decidir obedecendo às variações de volume, qual o maior volume com o qual você pode tocar a última música do espetáculo, de forma tal que em nenhum momento você ultrapasse um dado volume limite.

Por exemplo, suponha que você vá tocar três músicas e que, da primeira para a segunda música o volume deva variar 4 unidades e da segunda para a terceira música, o volume deva variar 7 unidades. Se o volume com o qual você vai tocar a primeira música é 5 e o volume máximo permitido é 9, há duas opções possíveis:

- tocar a primeira música com volume 5, a segunda música com volume $5 + 4 = 9$ e a terceira música com volume $9 - 7 = 2$;
- tocar a primeira música com volume 5, a segunda música com volume $5 - 4 = 1$ e a terceira música com volume $1 + 7 = 8$.

Dentre essas duas opções, a segunda é a que maximiza o volume com o qual a última música é tocada, e é ela que você vai usar.

2 Formato de Entrada

A entrada é composta de múltiplos casos de teste. Cada caso de teste é fornecido em duas linhas. A primeira linha contém 3 inteiros: N, o número de músicas no show, $volumeInicial$, o volume com o qual a primeira música será tocada, e $volumeLimite$, o maior volume permitido. Essas variáveis obedecem as seguintes restrições:

- $2 < N < 51$
- $0 < volumeInicial \leq volumeLimite \leq 10^3$

A segunda linha de cada caso de teste contém $N - 1$ inteiros que representam as alterações no volume de uma música para a próxima. O primeiro inteiro dessa linha representa a alteração da primeira para a segunda música. O segundo representa a alteração da segunda para a terceira música, e assim por diante. Todos esses valores estão entre 1 e $volumeLimite$, inclusive.

A entrada deve ser lida de um arquivo que contem os dados conforme o padrão informado acima.

3 Formato de saída

Para cada caso de teste, produza uma única linha na saída, contendo um único inteiro: o maior volume com o qual a última música pode ser tocada. Note que, em alguns casos, pode não ser possível tocar a última música com nenhum volume válido. Nesses casos, imprima -1.

4 Exemplos

Entrada Exemplo 1:

```
4 5 10
5 3 7
```

Saída Exemplo 1:

```
10
```

A entrada e o resultado acima significam que você abaixou o volume da guitarra para o nível 0(-5), na música seguinte aumentou para 3(+3) e na última música aumentou para 10(+7).

Entrada Exemplo 2:

```
5 8 20
15 2 9 10
```

Saída Exemplo 2:

```
-1
```

A entrada e o resultado acima significam que aumentar ou diminuir o volume inicial em 15 intervalos leva a um nível de volume inválido.

5 O que deve ser feito

1. Apresente uma solução de força bruta para o problema e implemente;
2. O problema tem subestrutura ótima? Se sim, prove.
3. O problema tem propriedade gulosa? Se sim, apresente e implemente uma solução gulosa e prove que ela é ótima.
4. Há sobreposição de subproblemas? Prove. Se sim, apresente e implemente uma solução utilizando programação dinâmica. Se não, prove que tal algoritmo não existe.
5. Realize uma avaliação experimental comparando as soluções apresentadas para o problema.

Formato documentação:

- Apresente a definição do problema;
- Explique as abordagens para solução do problema;
- Explique as decisões e módulos de sua implementação;
- Avaliação experimental comparando as soluções apresentadas para o problema;
- Faça uma análise de complexidade(tempo e espaço) das soluções apresentadas e justifique-a;
- Analise e discuta os resultados encontrados;
- Use o *template* de documentação presente no minha.ufmg;
- A documentação não pode exceder 10 páginas;

Código:

1. O código fonte do trabalho deve ser submetido para compilação e execução em ambiente Linux, tendo como padrão os computadores dos laboratórios de graduação do DCC;
2. Deve ser escrito na linguagem C (trabalhos implementados em outras linguagens como C++/Java/Python e outras não serão aceitos);
3. As estruturas de dados devem ser alocadas dinamicamente e o código ser modularizado (ou seja, dividido em múltiplos arquivos fonte e fazendo uso de arquivos cabeçalho - .h);
4. O utilitário Make deve ser utilizado para compilar e executar o programa;
5. A saída deve ser impressa no console do terminal seguindo estritamente o formato da especificação, caso contrário o resultado será considerado errado;
6. O arquivo executável deve ser chamado de **tp3_sigla** e deve receber como parâmetro apenas o nome do arquivo de entrada de dados. A *sigla* representa as possíveis soluções para o problema que são: força bruta (fb), algoritmo guloso (ag) e programação dinâmica (pd). Exemplo: tp3_fb, tp3_ag, tp3_pd. **Não serão aceitos outros nomes de executáveis além dos mencionados**.
7. Faça seu código de forma legível;

Entrega:

- Data de entrega : 24/05/2011 .
- Submissão: a documentação e o código do trabalho devem ser submetidos ao minha.ufmg. Para isso compacte os dois (formato tar.gz) e faça a submissão. Teste seu arquivo compactado antes de enviá-lo.
- A documentação também deve ser entregue impressa na secretaria do DCC. Não coloque nos escaninhos dos professores. A documentação deve ser entregue para a secretária e então colocada no envelope de AEDS3.
- Você receberá um e-mail agendando a entrevista do trabalho.

Distribuição dos pontos:

- Execução: 50%
- Documentação: 50%

Será adotado média harmônica entre a pontuação obtida na execução e na documentação do TP, o que implica em valor zero caso alguma das partes não seja apresentada.