Trabalho Prático 3: Legado da Copa

Algoritmos e Estruturas de Dados III – 2017/1
Entrega: 04/07/2017

1 Introdução

A Rua Pai no Bar é uma das ruas mais movimentadas em épocas comemorativas, esta rua possui vários bares e as pessoas não se incomodam muito com o barulho até de madrugada (exceto quando ocorre alguma desilusão amorosa e a pessoa que sofreu aumenta muito o volume do jukebox ao som de Telefone Mudo).

Os donos dos bares também residem nesta rua e, por serem pessoas muito desconfiadas, suas casas ficam sempre do lado da rua oposto ao bar (facilitando assim avistar movimentação estranha em sua casa ou bar).

Com a Copa na Rússia chegando os moradores da rua querem manter sua tradição de pendurar bandeirolas enfeitando a rua e assim dar alegria pro povo.

A crise impediu que os moradores conseguissem orçamento necessário para comprar novas bandeirolas, visto que as bandeirolas antigas já estão muito gastas. Todos ficaram muito tristes com o fato, mas foi aí que Sales, curiosamente conhecido por suas habilidades em imitar animais, sugeriu que fossem usadas as bandeirolas que os donos dos bares usam nos seus estabelecimentos.

Os donos dos bares também são moradores da rua e respeitam a tradição local. Eles concordaram com a ideia pois a rua enfeitada atrai movimento e consequentemente clientes. Eles concordaram em emprestar suas respectivas bandeirolas para enfeitar a rua com as condições a seguir:

- Cada bar vai contribuir com uma linha de bandeirola;
- As bandeirolas serão penduradas da seguinte maneira: Uma ponta da linha no bar e a outra ponta na casa do dono do bar.
- As linhas não podem se cruzar.

Cada bar não necessariamente fica em frente a casa do seu dono, um bar pode estar na esquina e a casa do dono na esquina oposta. A casa do dono do bar pode estar em qualquer lugar desde que seja do outro lado da rua onde está o bar.

Um dos moradores, Sami, fez uma tabela de duas colunas onde a primeira indica o número do bar e a segunda indica o número da residência do proprietário do bar. Você deve assumir que em um lado da rua temos apenas números ímpares e do outro lado da rua apenas números pares e que os números seguem em ordem crescente em ambos os lados da rua e na mesma direção (da mesma maneira que estamos acostumados).

Ajude os moradores desta rua a pendurarem o maior número de linhas de bandeirola possível respeitando as condições dadas acima.

2 Entrada e saída

Entrada A entrada começa com uma linha contendo 1 caractere no conjunto g, d, b, indicando se você deve usar um algoritmo guloso, programação dinâmica ou força bruta respectivamente.

A segunda linha contém 1 número natural positivo N representando o número de bares.

As próximas N linhas representam pares de números naturais positivos x, y onde x é um número do bar e y é o número da residência do proprietário do bar número x.

Exemplo de entrada

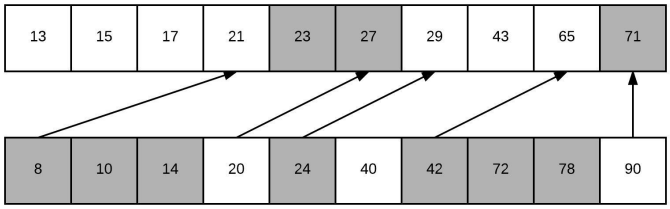
```
d
10
8  21
10  65
14  13
23  40
24  29
27  20
42  43
71  90
72  15
78  17
```

Saída A saída deve ser um número que representa a quantidade máxima de bandeirolas que podem ser penduradas na rua.

Exemplo de saída

5

O resultado é obtido ao utilizar as bandeirolas dos bares 8, 24, 27, 42 e 71. A figura abaixo representa como as ligações de bar/casas devem ser feitas. Os bares são representados pela área cinza e as casas em branco.



Observe que as linhas de bandeirolas não se cruzam

3 O que deve ser entregue

Deverá ser submetido um arquivo **.zip** contendo somente uma pasta chamada `tp3` e dentro desta deverá ter: (i) Documentação **em formato PDF** e (ii) Implementação.

Documentação Poderá ter no máximo 10 páginas e deverá seguir tanto os critérios de avaliação discutidos na Seção 4.1, bem como as diretrizes sobre a elaboração de documentações disponibilizadas no *moodle*. Além disso, a documentação deverá conter análise experimental validando as complexidades de tempo e espaço.

Implementação Código fonte do seu TP (*.c* e *.h*), com soluções baseadas em **Programação Dinâmica**, **Algoritmos Gulosos** e **Força Bruta**.

Makefile Inclua um *makefile* na submissão que permita compilar o trabalho. É obrigatório o uso das *flags*: **-Wall -Wextra -Werror -std=c99 -pedantic** na compilação.

4 Avaliação

Eis uma lista **não exaustiva** dos critérios de avaliação que serão utilizados.

4.1 Documentação

Introdução Inclua uma breve explicação do problema que está sendo resolvido no seu trabalho e um resumo da sua solução.

Solução do Problema Você deve descrever cada solução do problema de maneira clara e precisa, detalhando e justificando os algoritmos e estruturas de dados utilizados. Para tal, artifícios como pseudo-códigos, exemplos ou diagramas podem ser úteis. Note que documentar uma solução não é o mesmo que documentar seu código. **Não** é necessário incluir trechos de código em sua documentação nem mostrar detalhes de sua implementação, exceto quando estes influenciem o seu algoritmo principal, o que se torna interessante. A solução gulosa não precisa ser ótima, mas a solução por programação dinâmica e força bruta sim. Da mesma maneira, a solução por força bruta não precisa ser eficiente, mas a solução gulosa e solução por programação dinâmica sim.

Análise de Complexidade Inclua uma análise de complexidade de tempo e espaço dos principais algoritmos e estrutura de dados utilizados. Cada complexidade apresentada deverá ser devidamente **justificada** para que seja aceita. Faça uma análise para cada tipo de algoritmo implementado.

Avaliação Experimental Sua documentação deve incluir os resultados de experimentos que avaliem o tempo de execução de seu código em função de características da entrada. Cabe a você gerar entradas para esses experimentos. Para tal, um gráfico mostrando o tempo de execução em função do tamanho da entrada pode ser interessante. Você também deve interpretar os resultados obtidos. Comente sobre cada gráfico ou tabela que você apresentar mostrando o que é possível concluir a partir dele. Procure fazer uma análise de desempenho dos algoritmos implementados explicitando o compromisso entre custo assintótico e otimalidade da solução.

4.2 Implementação

Linguagem & Ambiente O seu programa deverá ser implementado na linguagem **C** e poderá fazer uso de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de outras bibliotecas que não a padrão serão zerados. Além disso, certifique-se que seu código compile e funcione corretamente nas máquinas **Linux** dos laboratórios do DCC.

Casos de teste A sua implementação passará por um processo de correção automatizado, portanto, o formato da saída do seu programa deve ser idêntico aquele descrito nessa especificação. Saídas com qualquer divergência serão consideradas erradas, mesmo que as divergências sejam *whitespaces*. e.g. espaços, *tabs*, quebras de linha, etc. Para auxilia-lo na depuração do seu código, será fornecido um pequeno, **não-exaustivo**, conjunto de entradas e suas respectivas saídas. É seu dever certificar-se que seu código funciona corretamente para qualquer entrada válida.

Alocação Dinâmica Algoritmos e estruturas de dados deverão fazer uso de memória alocada dinamicamente (`malloc()` ou `calloc()`). Certifique-se que seu programa utiliza essas regiões de memória corretamente, pois os monitores penalizarão implementações que realizam *out-of-bounds access* e que tenham vazamento de memória (não desalocar memoria dinâmica).
A alocação dinâmica deverá fazer uso das funções malloc() ou calloc() da biblioteca padrão C, bem como liberar tudo o que for alocado utilizando free(), para gerenciar o uso da memória. **DICA:** Utilize `valgrind` antes de submeter o seu TP.

Qualidade do código Seu código também será avaliado no quesito de legibilidade, dando atenção, porém não limitando-se, aos seguintes items: (i) **INDENTAÇÃO**; (ii) nomes de variável e função descritivos e claros; (iii) Modularização adequada; (iv) Comentários dentro de funções, explicando o que certos trechos mais complicados fazem; (v) Comentários fora de funções, explicando, em alto-nível, o que as funções mais importantes fazem; (vi) funções concisas que desempenham somente uma tarefa; (vii) **Proibido uso de variáveis globais**.

Atrasos Trabalhos poderão ser entregues após o prazo estabelecido, porém sujeitos a uma penalização regida pela seguinte fórmula:

$$\Delta_p = \frac{2^{d-1}}{0.32}\,\%$$

Por exemplo, se a nota dada pelo corretor for 70 e você entregou o TP com 4 dias corridos de atraso, sua penalização será de $\Delta_p = 25\%$ e, portanto, a sua nota final será: $N_f = 70 \cdot (1 - \Delta_p) = 52.2$. Note que a penalização é exponencial e 6 dias de atraso resultam em uma penalização de 100%.

5 Consideração Final

Assim como em todos os trabalhos dessa disciplina é estritamente proibida a copia parcial ou integra de códigos, seja da internet ou de colegas. Utilizaremos o algoritmo *MOSS* para detecção de plágio em trabalhos, seja honesto. Você não aprende nada copiando código de terceiros nem pedindo a outra pessoa que faça o trabalho por você. Se a cópia for detectada, sua nota será zerada e os professores serão informados para que as devidas providências sejam tomadas.

HAVE FUN!!!