

TP 03 - Trabalho Prático 03

Algoritmos I

Entrega: 05/07/2022

1 Objetivos do trabalho

O objetivo deste trabalho é modelar o problema computacional descrito a seguir utilizando uma estrutura de dados que permita resolvê-lo de forma eficiente com os algoritmos estudados nesta disciplina.

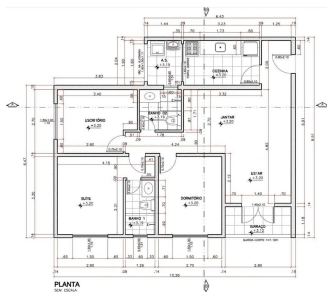
Serão fornecidos alguns casos de teste bem como a resposta esperada para que o aluno possa verificar a corretude de seu algoritmo. Não obstante, recomenda-se que o aluno crie casos de teste adicionais a fim de validar sua própria implementação.

O código-fonte da solução e uma documentação sucinta (relatório contendo não mais do que 5 páginas) deverão ser submetidos via *moodle* até a data limite de 05/07/2022. A especificação do conteúdo do relatório e linguagens de programação aceitas serão detalhadas nas seções subsequentes.

2 Reforma na casa da Vó

Apesar de toda tecnologia de encontros online é importante dar valor aos encontros familiares. Sua avó com certeza prefere encontros presenciais com toda a família e está reformando a casa dela para promover muitas reuniões no pós pandemia, pois ela sentia muita falta disso.

Como matriarca da família ela tinha um sonho de ter uma enorme mesa em casa, tão grande quanto seja possível, para que o máximo de pessoas possam se sentar junto a ela nas reuniões e jantares. Ela é uma vó coruja e acha alguns profissionais muito mais incompetentes que você, o neto(a) mais inteligente estudante de computação.



Sendo assim ela resolveu te promover o desafio da responsabilidade na escolha desta mesa que é o sonho dela. E você vai se esforçar com todas as suas forças para fazer a melhor escolha para sua querida vovózinha.

Não se desespere, ela forneceu tudo para que você resolva essa tarefa.

- A Arquiteta, mesmo contrariada com a desconfiança no trabalho dela, forneceu uma planta da casa informando quais espaços já estão ocupados e que, portanto, não podem conter a mesa..
- Sua Tia foi com sua vó na loja e listou o tamanho das várias mesas que sua vó gostou. Nesta lista contém o comprimento e a largura as opções de mesa.

Basta agora descobrir qual é a mesa de maior área que pode ser comprada e colocada na casa reformada! Vale lembrar que você pode rotacionar uma mesa em 90 graus, caso precise.

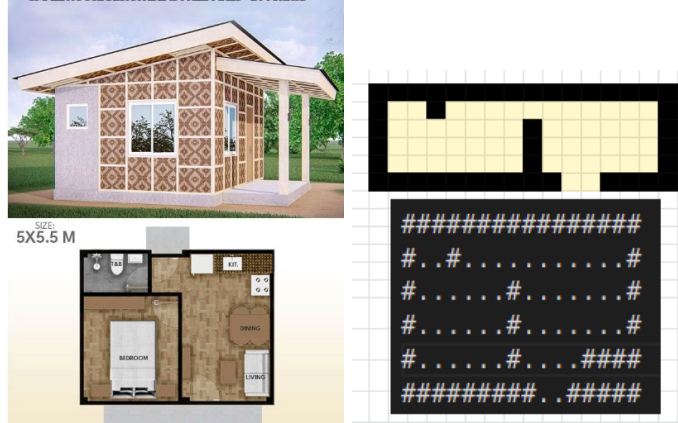


Figura 1: Diferentes formas de visão da planta de uma casa, do 3d ao console

3 O que fazer?

O **primeiro problema** é como converter a planta da casa, pois a *Arquiteta* que sua vó contratou te entregou uma imagem no formato que ela gosta de trabalhar. Mas para sua sorte isso foi fácil, existem sites que fazem esta conversão automaticamente.

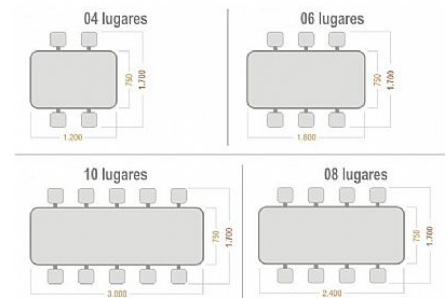
A título de exemplo, na Figura 4 (acima) temos um exemplo ilustrativo da conversão de formas de visão de uma casa:

- **Canto superior esquerdo:** A imagem 3d fornecida por um arquiteto.
- **Canto inferior esquerdo:** A planta baixa da mesma casa fornecida por um arquiteto.
- **Canto superior direito:** Representação ilustrativa dos espaços vazios e os espaços ocupados, por paredes ou objetos (Sem relação correta de proporção com a planta da mesma imagem).
- **Canto interior direito:** Representação de console que o site te fornece e perfeita para a nossa aplicação computacional.

O **segundo problema** é mais simples, (como visto ao lado) as mesas tem diferentes dimensões e ainda é preciso considerar o espaço das cadeiras para caber no espaço previsto. Para nossa facilidade sua tia já fez a medição das mesas definindo o espaço necessário que inclui a mesa e as cadeiras prevendo pessoas sentadas.

O **terceiro problema** também já foi simplificado e é relações das dimensões. Diremos que cada caractere da representação usado no mapa do console representa $1m^2$ (Hum metro Quadrado), logo se a casa for definida na entrada por 6×16 , significa que ela tem 6m por 16m e a dimensão da mesa será sempre feito de forma inteira, se a mesa tem 2×4 ela precisa do espaço de 2m por 4m.

Você percebeu que simplificamos e resolvemos os problemas 1, 2 e 3 que são relacionados à entrada mas agora vem a sua parte.



O **quarto problema é seu**: Você receberá a planta no formato de console junto com a lista de mesas e seu programa deverá:

- Analisar os "cômodos" da casa, ou seja, encontramos uma lista Q de todos os retângulos liberados (com apenas caracteres '.').
- Para cada mesa, verificamos se existe um retângulo liberado em Q que pode caber a mesa.

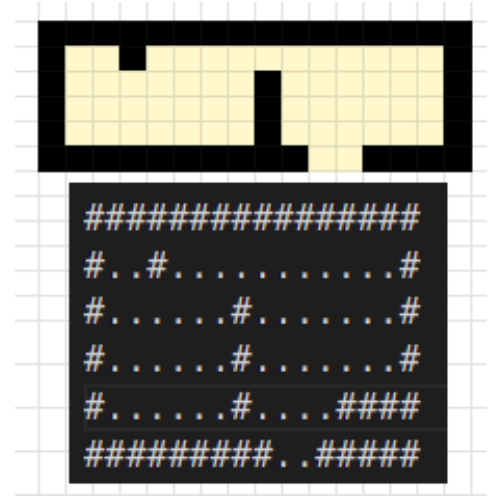
4 Exemplo do problema

A primeira casa de teste antes de usar o seu sistema com sua vó é um salão de 96m^2 $6\text{m} \times 16\text{m}$. Ela tem uma porta de entrada e 3 cômodos, uma sala conectada a um quarto e um banheiro no canto esquerdo. A planta desta casa foi convertida fornecida em forma de console como na figura do lado esquerdo. Para você não há distinção entre os cômodos, você deve localizar a maior mesa que cabe dentro deste espaço, mesmo que seja dentro do banheiro.

Junto com esta casa, foram cotadas 4 mesas:

3 X 3, 4 X 6, 15 X 2, 2 X 7.

- **3 X 3** (9m^2): Cabe! Mesa quadrada, cabe tanto na sala quanto no quarto.
- **4 X 6** (24m^2): Esta mesa retangular não cabe nos espaços descritos acima.
- **15 X 2** (30m^2): Esta mesa retangular não cabe nos espaços descritos acima.
- **2 X 7** (14m^2): Esta mesa cabe no primeiro comodo próximo à porta.



5 Arquivos de entrada

O programa terá como entrada na sua primeira linha linha contendo dois inteiros N e M separados por espaço, que são as dimensões da casa da avó. Em seguida, há N linhas contendo M caracteres descrevendo a planta arquitetônica. Um caractere . (ponto) representa um espaço vazio, enquanto que o caractere # representa um espaço preenchido (parede ou outro móvel).

A linha seguinte contém um inteiro K que representa o tamanho da lista de opções de mesa que sua avó pode comprar. As próximas K linhas contém, cada uma, dois inteiros C_i e L_i separados por espaço, representando o comprimento e largura da i -ésima mesa.

Restrições

$$1 \leq N, M \leq 1000$$

$$1 \leq K \leq 10^6$$

$$1 \leq C_i \leq \min(500, \max(N, M))$$

$$1 \leq L_i \leq \min(500, \max(N, M))$$

6 Saída

A saída é apenas uma linha contendo dois inteiros separados por espaço: Estes dois inteiros são o **comprimento e a largura** da mesa de maior área que cabe na casa da avó.

Em caso de **empate**, imprima as dimensões da mesa de maior largura. É garantido que sempre há pelo menos uma mesa que cabe na casa.

7 Exemplo Prático de Entrada e Saída

Arquivo contendo o mapa da casa e as mesas

```
6 16
#####
#.#.....#
#.....#.....#
#.....#.....#
#.....#.....####
#####..#####
4
3 3
4 6
15 2
2 7
```

Exemplo prático da saída

```
2 7
```

Arquivo contendo o mapa da casa e as mesas

```
13 32
#####
#.....##.....##.....#
#.....##.....##.....#
#.....##.....##.....#
#.....##.....#####..#
#..#####.....#.....#
#.....#.....#.....#
#..#####.....###.....#
#.....####.....###.....#
#.....##.....#.....###.....##
#.....##.....#.....#.....##
#.....##.....#.....#####
#####.#####
4
3 3
3 6
15 2
7 7
```

Exemplo prático da saída

```
7 7
```

8 Especificação das entregas

Você deve submeter um arquivo compacto (zip ou tar.gz) no formato **MATRICULA_NOME** via Moodle contendo:

- todos os arquivos de código-fonte implementados;
- sua documentação (arquivo pdf).
 - **ATENÇÃO:** O arquivo compactado NÃO deverá conter um arquivo *makefile*, haja visto que o comando "make" não está funcionando nas máquinas do laboratório.
 - **ATENÇÃO:** É de suma importância que a documentação contenha instruções expressas sobre a forma adequada de compilar o seu programa (Em particular, o comando ou sequência de comandos utilizado, exemplo: "g++ main.cpp").

Sua documentação deverá ser sucinta e conter não mais do que 5 páginas com o seguinte conteúdo obrigatório:

- Modelagem computacional do problema;

- estruturas de dados e algoritmos utilizados para resolver o problema (pseudo-código da solução implementada), bem como justificativa para tal escolha. Não transcreva trechos da código-fonte;
- análise de complexidade de tempo assintótica da solução proposta, devidamente justificada.

9 Implementação

9.1 Linguagem, Ambiente e Parâmetros

O seu programa deverá ser implementado na linguagem **C** ou **C++** e deverá fazer uso apenas de funções da biblioteca padrão da linguagem. Trabalhos que utilizem qualquer outra linguagem de programação e/ou que façam uso de bibliotecas que não a padrão não serão aceitos.

O aluno pode implementar seu programa em qualquer ambiente (Windows, Linux, MacOS, etc...), no entanto, deve garantir que seu código compile e rode nas máquinas do DCC (tigre.dcc.ufmg.br ou jaguar.dcc.ufmg.br), pois será neste ambiente que o TP será corrigido. Note que essas máquinas são acessíveis a todos os alunos do DCC com seu login e senha, podendo inclusive ser realizado acesso remoto via ssh. O aluno pode buscar informações no site do CRC (Centro de Recursos Computacionais) do DCC (<https://www.crc.dcc.ufmg.br/>).

O arquivo da entrada deve ser passado ao seu programa como entrada padrão, através da linha de comando (e.g., `$./tp03 < casoTeste01.txt`), e gerar o resultado também na saída padrão (não gerar saída em arquivo). Informações sobre entrada e saída padrão podem ser encontradas na página web a seguir: https://cplusplus.com/doc/tutorial/basic_io/.

ATENÇÃO: Não é necessário que o aluno implemente em ambiente Linux. Recomenda-se que o aluno teste seu código nas máquinas previamente especificadas, as quais serão utilizadas para correção do TP, a fim de conferir a funcionalidade e demais características do código.

ATENÇÃO: Falha em atender as especificações de entrada e saída descritas nessa seção resultará na rejeição do trabalho.

9.2 Testes automatizados

A sua implementação passará por um processo de correção automatizado, utilizando além dos casos de testes já disponibilizados, outros exclusivos criados para o processo de correção. O formato da saída de seu programa deve seguir a especificação apresentada nas seções anteriores. Saídas diferentes serão consideradas erro para o programa. O aluno deve certificar-se que seu programa execute corretamente para qualquer entrada válida do problema.

ATENÇÃO: O tempo máximo esperado para execução do programa, dado o tamanho máximo do problema definido em seções anteriores, é de 5 segundos.

9.3 Qualidade do código

Preze pela qualidade do código-fonte, mantendo-o organizado e comentado de modo a facilitar seu entendimento para correção. Caso alguma questão não esteja clara na documentação e no código fonte, a nota do trabalho pode ser penalizada.

10 Critérios para pontuação

A nota final do TP (NF) será composta por dois fatores: fator parcial de implementação (fpi) e fator parcial da documentação (npd). Os critérios adotados para pontuação dos fatores é explicado a seguir.

10.1 Fator parcial de implementação

Serão avaliados quatro aspectos da implementação da solução do problema, conforme a Tabela 1.

Aspecto	Sigla	Valores possíveis
Compilação no ambiente de correção	co	0 ou 1
Respostas corretas nos casos de teste de correção	ec	0 a 100%
Tempo de execução abaixo do limite	te	0 ou 1
Qualidade do código	qc	0 a 100 %

Tabela 1: Aspectos de avaliação da implementação da solução do problema

O fator parcial de implementação será calculado pela seguinte fórmula:

$$fpi = co \times (ec - 0,15 \times (1 - qc) - 0,15 \times (1 - te))$$

Caso o valor calculado do fator seja menor que zero, ele será considerado igual a zero.

10.2 Fator parcial da documentação

Serão avaliados quatro aspectos da documentação entregue pelo aluno, conforme a Tabela 2.

Aspecto	Sigla	Valores possíveis
Apresentação (formato, clareza, objetividade)	ap	0 a 100%
Modelagem computacional	mc	0 a 100%
Descrição da solução	ds	0 a 100%
Análise de complexidade de tempo assintótica	at	0 a 100 %

Tabela 2: Aspectos de avaliação da documentação

O fator parcial de documentação será calculado pela seguinte fórmula:

$$fpd = 0,4 \times mc + 0,4 \times ds + 0,2 \times at - 0,25 \times (1 - ap)$$

Caso o valor calculado do fator seja menor que zero, ele será considerado igual a zero.

10.3 Nota final do TP

A nota final do trabalho prático será obtida pela equação a seguir:

$$NF = 10 \times (0,6 \times fpi + 0,4 \times fpd)$$

É importante ressaltar que é obrigatória a entrega do código fonte da solução e documentação. Na ausência de um desses elementos, a nota do trabalho prático será considerada igual a zero, pois não haverá possibilidade de avaliar adequadamente o trabalho realizado.

Assim como em todos os trabalhos dessa disciplina é estritamente proibida a cópia parcial ou integral de código-fontes, seja da Internet ou de colegas. Se for identificado o plágio, o aluno terá a nota zerada e o professor será informado para que as medidas cabíveis sejam tomadas.

ATENÇÃO: Os alunos que submeterem os TPs com atraso, terão a nota final penalizada em termos percentuais de acordo com a seguinte regra: $2^{d-1}/0,16$ (onde d é a quantidade de dias úteis de atraso na entrega do TP)