

Estrutura de Dados e Algoritmos  
Projeto 04: Construção de Shopping  
Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



# 1 Contextualização

A empreiteira *DSNN Construtora* quer aproveitar uma sequência de prédios consecutivos em uma rua para criação de um shopping. Cada um dos prédios pode ser visto como um retângulo de largura 1 e altura  $h_i$ . Estes prédios não possuem separação entre eles, isto é, são geminados.

O shopping a ser utilizado no espaço dos prédios deve ser retangular, ou seja, ele deve possuir uma altura  $h_s$  e largura  $b_s$  e, para alcançar este formato, a construtora pode demolir alguns andares superiores de prédios que ela desejar. Cada andar possui altura 1. Para maximizar o lucro, a área do Shopping deve ser a maior possível.

Tomando como exemplo um cenário em que tenhamos 5 prédios, com alturas  $h = (1, 2, 3, 4, 5)$ , a área máxima do shopping a ser construído é 9, conforme a Figura 1.

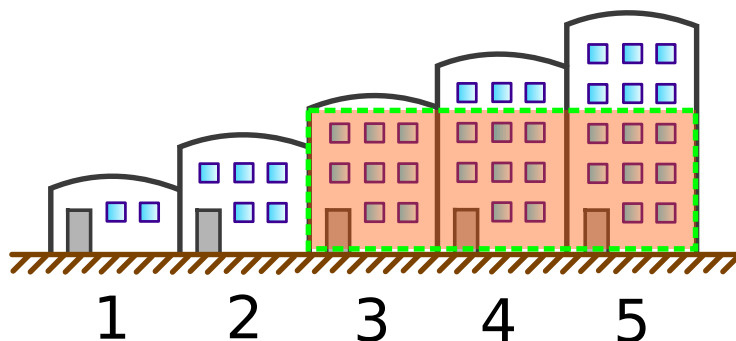


Figura 1: Construção de um shopping retangular de área 9.

Neste exemplo, um andar do quarto prédio é demolido e dois andares do quinto prédio é demolido, possibilitando a utilização de um espaço pelo Shopping de largura 3 (a partir do terceiro prédio) e altura 3.

## 2 Especificação

O projeto deverá ser executado através da linguagem C.

A entrada deve ser lida da entrada padrão (`stdin`), enquanto a saída deverá ser impressa na saída padrão (`stdout`).

O programa deverá obedecer rigorosamente o formato de saída especificada, pois parte da correção será automatizada.

Obrigatoriamente a estrutura de dados “Pilha” deverá ser utilizada para resolver este problema.

### 2.1 Entrada

A primeira linha possui um inteiro  $n$  ( $1 \leq n \leq 10^5$ ) indicando o número de prédios.

A próxima linha possui  $n$  inteiros, separados por espaço, indicando a altura  $h_i$  ( $1 \leq h_i \leq 10^7$ ) de cada prédio.

## 2.2 Saída

Seu programa deverá imprimir em uma única linha a área máxima possível do Shopping.

## 3 Exemplos

- Entrada

```
5
1 2 3 4 5
```

- Saída:

```
9
```

- Entrada

```
5
3 1 2 1 3
```

- Saída:

```
5
```

- Entrada

```
5
5 4 1 4 5
```

- Saída:

```
8
```

### 3.1 Limites de Tempo e Memória

Para cada caso de teste, será permitido a execução do programa por apenas 1 segundo com utilização máxima de 256 MB de memória. Caso o programa leve mais tempo ou memória do que isso, considerar-se-á que o algoritmo empregado foi ineficiente.

### 3.2 Documentação

Junto do(s) código(s) necessário(s) para resolver o problema, deverá ser disponibilizado um arquivo README, identificando o autor do trabalho e especificando as instruções para compilação e execução do(s) código(s).

### 3.3 Critérios de Correção

Fazem partes dos critérios de correção:

- Eficiência do programa.
- Utilização de estruturas de dados adequadas.
- Documentação: além do arquivo README, o código deve estar bem documentado.
- Legibilidade.

### 3.4 Ambiente de Correção

Os projetos serão corrigidos em uma máquina com sistema GNU/Linux e compilador `gcc 10.2.0`.

Trabalhos que não compilarem não serão avaliados.

## 4 Considerações

- Este projeto deve ser executado individualmente.
- Os trabalhos que incidirem plágio serão avaliados automaticamente com nota 0 para os envolvidos. Medidas disciplinares também serão tomadas.
- O trabalho deve ser entregue dentro de uma pasta zipada com a devida identificação do(s) aluno(s) através da sala de aula virtual da disciplina na data estipulada no ambiente.