

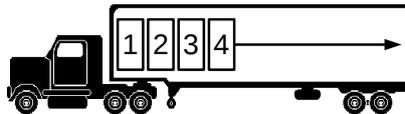
Instituto de Computação – UNICAMP
MC202 – Estrutura de dados
Prof. Lehilton – Turmas G e H
[MC202GH] Tarefa Extra

Instruções

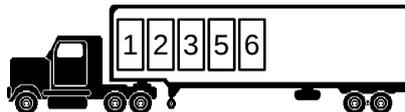
- Essa tarefa é opcional!
- Resolva **todas** as quatro questões completamente.
- As respostas devem ser enviadas como código-fonte em um **único** arquivo de texto com o formato XXXXXX.c onde XXXXXX é o número do RA. O arquivo deve compilar sem erros ou avisos.
- A questão 4 deverá ser respondida como um comentário do arquivo .c, e deve ser devidamente identificada.
- O arquivo deverá ser anexado em o e-mail enviado ao professor até o dia 18/9 contendo obrigatoriamente como assunto: [MC202GH] Tarefa Extra.
- Você pode discutir sobre a tarefa com colegas, mas cada um deverá responder individualmente. Não serão aceitas cópias.

EMPRESA TRANSPORTADORA

Um caminhão deve sair de um depósito inicial (identificado por 0), entregar diversos pedidos em outros depósitos (identificados por números 1,2,...) e voltar ao depósito inicial. Cada pedido consiste de uma grande caixa padronizada que é colocada dentro do caminhão. Acontece que devido ao tamanho das caixas, elas só podem ser retiradas na ordem inversa daquela em que foram colocadas. Na imagem a seguir, a ordem em que as caixas foram colocadas foi 1, 2, 3 e 4, ou seja, o caminhão deverá ir primeiro ao depósito 4.



Chegando a um depósito, a caixa correspondente é retirada e outros pedidos podem ser incluídos. Na figura a seguir, após o pedido 4 ser retirado, novos pedidos 5 e 6 foram incluídos. Assim, o próximo depósito a ser visitado é o 6.



No exemplo, se nenhum outro pedido for incluído após o 6, então a rota completa percorrida pelo caminhão será $0 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$.

Questão 1. O tempo de percurso entre dois depósitos é guardado em uma matriz T armazenada em um **vetor de vetores de inteiros**, onde $T_{i,j}$ é a duração em minutos para ir da localidade i à j .

a) Escreva uma função que leia do teclado:

- na primeira linha: o número de localidades n ;
- nas próximas n linhas: os valores correspondentes às linhas de T separados por espaço.

O protótipo deverá ser: `int **ler_matriz(int *n)`. **Lembre-se de alocar espaço para todos os vetores!**

b) Escreva uma função **recursiva** que receba a matriz T e verifique se T é simétrica, isso é, se o tempo de percurso de i até j é o mesmo de j até i , para todos depósitos i e j .

Cada pedido p é formado por um depósito i que identifica o pedido e um prazo de retorno t (por simplicidade, esse prazo é representado por um número de minutos a partir do início do expediente). Esse prazo significa que o caminhão que transporta o pedido p deve **voltar ao depósito inicial** antes do tempo t .

Questão 2. Por causa dos prazos, além do conjunto de pedidos colocados em um caminhão, é importante guardar o **menor** prazo de entrega desses pedidos. Considere um tipo abstrato de dados (TAD) “**pilha de**

pedidos com menor prazo” com as operações listadas a seguir; cada operação nesse TAD executa apenas $O(1)$ instruções:

- iniciar pilha; destruir pilha; verificar se pilha é vazia;
- empilhar pedido; desempilhar pedido;
- obter o número do depósito no topo da pilha;
- obter o menor prazo.

a) Descreva a *interface* da estrutura de dados abstrata.

b) Implemente **TODAS** as operações.

Questão 3. O gerente da transportadora decidiu usar o seguinte algoritmo para encher um caminhão e seguir a rota:

1. começar no depósito 0;
2. no depósito corrente, ler cada novo pedido em ordem:
 - calcular a duração da nova rota com a inclusão do pedido,
 - se a nova rota não prejudicar prazo o dos itens no caminhão, empilhar pedido;
3. se houver pedidos no caminhão, ir para o próximo depósito, desempilhe e continue no passo 2;
4. se não houver mais pedidos, retornar ao depósito 0.

a) Usando a interface do TAD definido na Questão 2(a), escreva uma função que implemente o *passo 2* do algoritmo. A função receberá: a pilha com os pedidos atual; o tempo total da rota atual; o identificador do depósito corrente e a matriz T . A função devolverá: a duração da nova rota. A função deverá imprimir o número do depósito corrente para o usuário e, em seguida, ler do teclado a sequência de pedidos do depósito corrente:

- na primeira linha: o número de pedidos m ;
- nas próximas m linhas: o número do depósito destino e o prazo de retorno de um pedido.

b) Escreva uma função que imprima a rota realizada por um caminhão de acordo com o algoritmo definido pelo gerente.

Questão 4. Lembrando que escrevemos $f(n) = O(g(n))$ se existirem números positivos c e n_0 tais que $f(n) \leq cg(n)$ para todo $n \geq n_0$, responda:

a) Mostre usando a definição: $2n + 3 \log n^4 = O(n)$.

b) Seja $T(n)$ o número de multiplicações realizadas pela função a seguir. Prove que $T(n) = O(\log_2 n)$.

Dica: para um argumento rigoroso, mostre por indução que $T(n) \leq 2 \log_2 n$ para todo $n \geq 1$.

```
float potencia(float x, int n) {
    float r;
    if (n == 1) return x;
    if (n % 2 == 1) {
        r = potencia(x, (n-1)/2);
        return x*r*r;
    } else {
        r = potencia(x, n/2);
        return r*r;
    }
}
```

c) Assumindo que o pivô de particionamento escolhido é sempre o último elemento, dê um exemplo em que o *Quick-Sort* realiza $\Omega(n^2)$ comparações. Justifique informalmente, mas de forma clara.