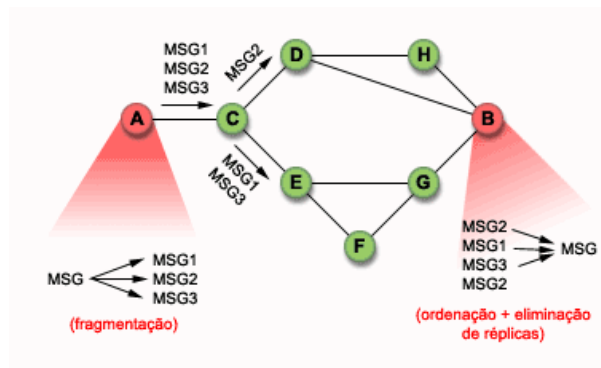


Avaliação Parcial 2

Redes de comunicação de dados. A maioria dos protocolos de comunicação fragmenta as mensagens em pacotes que são numerados e enviados através da rede. Não há nenhuma garantia de que os pacotes chegarão ao seu destino na ordem em que foram enviados pelo remetente. Além disso, pode haver (por causa da perda de mensagens de confirmação) a duplicação de alguns pacotes que chegam ao destinatário. A fim de reconstituir a mensagem original, o destinatário precisa descartar as réplicas de pacotes recebidos e ordená-los. O uso de árvores binárias possibilita a implementação de um algoritmo eficiente para resolver ambos os problemas.



Algoritmo.

- O primeiro número da lista é colocado num nó que é a raiz da árvore, com as sub-árvores esquerda e direita vazias.
- Cada número sucessivo é comparado com o número da raiz.
 - Se for igual, achou uma réplica (descarta-se o número).
 - Se é menor, examinamos a sub-árvore esquerda; se é maior, examinamos a sub-árvore direita.
 - Se a sub-árvore é vazia, o número não está duplicado e é colocado em um novo nó naquela posição da árvore.
 - Se a sub-árvore não for vazia, comparamos o número com o conteúdo da raiz da sub-árvore e o processo é então repetido para a sub-árvore.

Uma vez construída a árvore, basta percorrê-la em ordem simétrica (*in-ordem*) para obter os seus elementos em ordem crescente.

Questão 1.

2 P.

Tendo por base o que foi descrito acima, crie uma estrutura `struct mensagem` com os membros `int pos` e `char info`. Crie uma função que recebe uma string como parâmetro e retorna um vetor do tipo `struct mensagem`. A função deve preencher as estruturas do vetor de mensagens com os caracteres da string e as suas posições. Exemplo:

```
struct mensagem m[3];
char palavra[3] = "01a";
m[0].pos = 0;
m[0].info = palavra[0];
```

```
m[1].pos = 1;
m[1].info = palavra[1];
m[2].pos = 2;
m[2].info = palavra[2];
```

Questão 2.

2 P.

Crie uma função para enviar as mensagens que retorna um vetor de mensagens em ordem aleatória e com mensagens duplicadas. Selecione aleatoriamente posições no vetor de mensagens para serem duplicadas.

Questão 3.

2 P.

Crie uma função que recebe um vetor de mensagens e retorna uma árvore binária preenchida seguindo as instruções do algoritmo no texto acima.

Questão 4.

2 P.

Crie uma função que recebe como parâmetro uma árvore binária com as mensagens e a percorre em ordem simétrica, a medida que um novo nó é avaliado este deve ser colocado em uma fila de mensagens. Terminado o percurso pela árvore, retorne um ponteiro para a fila de mensagens.

Questão 5.

2 P.

Na função `main` use a função da questão anterior para recuperar uma fila de mensagens com a mensagem original completa e na ordem correta, e então imprima a mensagem.