SCC0201/501 (Introdução à Ciência de Computação II) — Prof. Moacir P. Ponti Junior

Trabalho 1

Implemente sua atividade sozinho sem compartilhar, olhar código de seus colegas, ou buscar na Internet. Procure usar apenas os conceitos já vistos nas aulas.

Manipulando conjuntos ordenados

Definiremos um **conjunto ordenado** como sendo uma lista ordenada de números inteiros não duplicados. Assim, a seguinte lista de inteiros:

`5, 9, 16, 9, 3, 0, 5, 5, 15, 16`

Gera um conjunto ordenado (em ordem crescente) como segue:

`0, 3, 5, 9, 15, 16`

No computador um conjunto ordenado é, na verdade um arranjo total ou parcialmente preenchido. E, portanto, consiste de:

1. Um arranjo para armazenar os números no conjunto
2. Uma variável inteira utilizada para guardar a informação do tamanho utilizado do arranjo.

É possível realizar diversas operações para manipular esses conjuntos, como união, intersecção, diferença, entre outras. A **intersecção** entre dois conjuntos ordenados é definida como um novo conjunto contendo os inteiros em comum a ambos os conjuntos. Considere, por exemplo os dois conjuntos abaixo:

`A = 1, 5, 7, 8, 12, 20`

`B = 2, 4, 6, 8, 10, 12, 19, 20, 21`

A intersecção entre entre os conjuntos, `A` $\cap$ `B` é:

`8, 12, 20`

A **união** entre dois conjuntos ordenados é definida como um novo conjunto contendo todos os inteiros sem repetição contidos em ambos os conjuntos. Considere novamente os conjuntos `A` e `B` acima, a união: `A` $\cup$ `B` é:

`1, 2, 4, 5, 6, 7, 8, 10, 12, 19, 20, 21`

A **diferença** `A` $-$ `B` é um conjunto ordenado dos números presentes em `A` e que não estejam presentes também em `B`. Considere novamente os conjuntos `A` e `B` acima, a diferença `A` $-$ `B` é:

`1, 5, 7`

Tarefa

Desenvolva um programa que receba como entrada 2 listas de números inteiros (L_1 e L_2), gere 2 conjuntos ordenados correspondentes a cada uma delas (C_1 e C_2), e gere como saída:

1. a intersecção entre os dois conjuntos: $C_1 \cap C_2$,
2. a união dos dois conjuntos: $C_1 \cup C_2$,
3. a diferença entre o primeiro e o segundo conjunto: $C_1 - C_2$

Você deverá primeiramente ordenar ambas as listas e eliminar os números duplicados, gerando os dois conjuntos. A seguir, deverá realizar as operações e gerar como saída os conjuntos derivados das 3 operações definidas acima (intersecção, união e diferença).

O algoritmo de ordenação a ser utilizado deverá ser o **cocktail sort**. Esse algoritmo é uma variante do péssimo algoritmo bubble sort. O cocktail sort ordena em ambas as direções em cada passada pela lista.

Considere uma lista de tamanho n com posições de 0 a $n-1$: enquanto o bubble sort a cada passada move o maior elemento para o final da lista, volta para o início e repete essa operação, o cocktail sort começa do primeiro elemento, realiza comparações dois a dois e move o **maior** elemento para o final da lista. A seguir, começando do elemento $n-2$ e indo até o elemento inicial o algoritmo compara dois a dois e move o **menor** elemento para o início da lista.

Procure mais informações sobre esse algoritmo para entender seu funcionamento e implemente-o como uma função em seu código-fonte para utilizar na criação dos conjuntos.

Entrada e saída

A entrada será composta de:

- dois números inteiros n e m, sendo que: $1 \leq n, m \leq 30$, onde n representa o número de elementos na lista L_1 e m representa o número de elementos na lista L_2.
- duas sequências de inteiros, a primeira de tamanho n e a segunda de tamanho m.

A entrada de n e m deverá ser feita na mesma linha. A seguir, cada sequência deve ser feita em uma linha separada.

A saída será composta de três sequências de inteiros (uma em cada linha cada número sendo separado por um espaço) relativas à operações sobre os conjuntos gerados a partir das listas L_1 e L_2, na sequência:

1. a intersecção entre os conjuntos,
2. a união dos conjuntos,
3. a diferença entre o primeiro e o segundo conjunto.

Exemplo de entrada

```
6 9
1 3 3 1 5 4
8 10 3 1 8 8 9 0 7
```

Exemplo de saída

```
1 3
0 1 3 4 5 7 8 9 10
4 5
```

Instruções

O projeto será avaliado principalmente levando em consideração:

1. Processamento correto das entradas e saídas do programa
2. Realização das tarefas descritas
3. Bom uso das técnicas de programação
4. Boa endentação e uso de comentários no código

Restrições:

- **Não** deverá ser utilizada qualquer variável global.
- As seguintes funções deverão **obrigatoriamente** ser implementadas e utilizadas:
  1. `int gerarconjunto(int L[], int tamL, int C[])` – deve receber como parâmetro uma lista `L` de tamanho `tamL`, modificar o arranjo `C` de forma que esse receba o conjunto ordenado calculado a partir da lista `L`, e retornar o número de elementos no conjunto ordenado `C` gerado.
  2. `void cocktailsort(int A[], int tamA)` – deve receber como parâmetro um arranjo de inteiros `A` de tamanho `tamA`, e utilizar o algoritmo cocktail sort para modificar o arranjo `A` de forma que esse contenha os elementos rearranjados em ordem crescente.
- Você poderá criar outras funções se quiser ou precisar — lembre-se de tornar a função criada útil para os propósitos de reuso e abstração, e de comentá-la corretamente.
- Você deverá realmente implementar o cocktail sort, sem utilizar bibliotecas com funções prontas.

ATENÇÃO: o projeto deverá ser entregue apenas pelo SQTPM no formato **c**, escolhendo a opção `Trabalho1`. Há dois casos de teste para serem baixados na página da disciplina. O sistema receberá trabalhos **apenas** entre os dias 09/09 as 0h00 e 13/09 as 23h59. Não serão aceitos trabalhos com atraso.