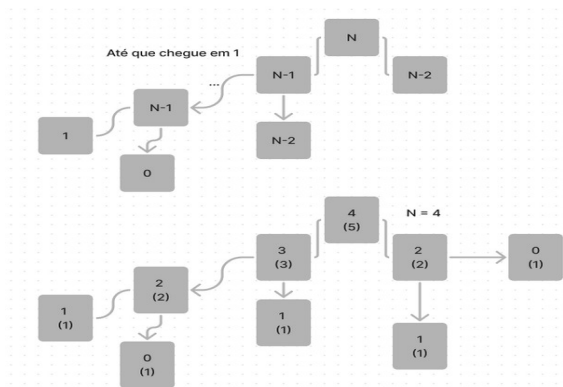


Parte IV: PROGRAMAÇÃO DIRETA E RECURSIVA

1. Mostre o que se passa no método a seguir desenhando em diagrama de chamadas a sua execução. Verifique a sua complexidade computacional.

```
1. procedure Robot(n)
2. If n in [0,1] then return(1)
3. else return(Robot(n-1)+Robot(n-2));
4. end Robot
```

solução:



Assim, temos uma complexidade computacional de $O(2^n)$, e para a memória $2n$.

2. Escreva a versão não recursiva para o método a seguir, e indique o que ele faz:

```
1. procedure XMC(m, n)
2. If m < n Then Troca(m, n) // faz m ser o maior
3. If n=0 Then
4. return(m)
5. r:=m mod n
6. return(XMC(n, r))
7. end XMC
```

solução:

```
procedure XMC!(m, n)
  if m < n Then Troca(m, n) // Faz m ser o maior como no algoritmo de cima
  while (n != 0) do
    z:= m mod n
    m:= n
    n:= z
  return (m)
end XMC!
```

Esse método, assim como o outro, calcula o XMC entre dois números, se diferenciando apenas no modo de se calcular, que antes era na forma recursiva, agora se utiliza de um laço while que faz a mesma função, porém sem o custo de uma chamada recursiva.

3. Escreva um algoritmo recursivo para o seguinte problema: “Um robô pode caminhar dando passos de 1 ou 2 metros, o método deve mostrar o número de maneiras que o robô pode andar n metros, onde $n \geq 1$ ”. Também responda, este algoritmo atende para todo $n \in \mathbb{N}^*$?

solução:

```
procedure movimentoRobo(n)
  if n = 1 Then return (1)
  return (movimentoRobo(n-1) + (movimentoRobo(n-2)))
end movimentoRobo
```

obs: Para todo $n \geq 1$.

4. Escreva um algoritmo recursivo e outro não recursivo que resolva o problema de Lucas, para um dado $n \geq 1$, onde n é o número de pinos de mesma cor, e $n \in \mathbb{N}^*$.

5. Escreva um algoritmo recursivo que inverta uma dada sequência S de números inteiros. Mostre usando indução matemática que seu algoritmo é correto.

Solução:

```

procedure inverte(n, x, y)
  if x < y Then
    aux:=  $n_x$ 
     $n_x := n_y$ 
     $n_y := aux$ 
    inverte (n, x+1, y-1)
end inverte

procedure inverteSeq(n)
   $x := n_{\text{inicioVetor}}$ 
   $y := n_{\text{fimVetor}}$ 
  inverte ( n, x, y)
end inverteSeq

```

prova:

Para $n = 1$, temos que o seu inverso será ele mesmo. Logo para $n = k$ temos a mesma validação, agora para $k+1$ temos que o passo de inversão trocava $(k+1) + 1$ por $(k+1) - 1$, que é igual a $k + 1$, até que o primeiro passe a ser maior que o segundo. Assim, todos os elementos serão invertidos até que o meio do conjunto seja alcançado, tendo invertido sua ordem.

C.Q.D.

6. Descreva a função de quantidade de iterações do método a seguir:

```

1. procedure RaizQ(x, r) // Cálculo da Raiz Quadrada
2. If  $|x-r^2| \leq \text{Tol}$  Then
3. return(r)
4. RaizQ(x, (x/r + r)/2);
5. end RaizQ

```

Solução:

A função que descreve a quantidade de iterações do algoritmo pode ser dado pela função:

$$f(x) = \frac{\frac{1}{2} * \log\left(\frac{x}{r} + r\right)^2}{\log 2}$$

Isso vai depender da tolerância passada, variando então o número de iterações.

7. Descreva a função genérica para calcular $f(x)=x^{n/m}$ com precisão ε dada, e onde $x \in \mathbb{R}$, $n, m \in \mathbb{N}$, $m \neq 0$:

Solução:

A função que calcula $f(x) = x^{\left(\frac{n}{m}\right)}$ seria de maneira similar a própria **RaizQ**:

```

procedure RaizN(x, r)
  while  $|x^n - r^m| \leq \varepsilon$  do
     $r := ((x^n / r^{(m-1)}) + (m - 1) r) / m$ 
  return ( r )
end RaizN

```

8. Um bêbado sai do bar à meia noite e meia e segue para a sua casa que fica a 280m de distância em linha reta. O bêbado dá passos aleatórios para frente e para trás de 1, 2 e 3m sem cair. Elabore um procedimento que calcula quantos passos o bêbado dará para chegar em casa. Se cada metro que ele vence à frente leva 1s e 2s para trás, qual o tempo que o bêbado levará em média para chegar em casa?

Solução:

```

procedure ChegarEmCasa()
  metros:=0
  tempo:=0
  passos:=0
  while metros < 280 do
    qtde:=Random(1,2,3)
    frenteTras:=Random(1,2)
    if frenteTras = 2 Then

```

```

        metros:= metros - qtde
    else Then
        metros:= metros + qtde
    tempos:=metros*frenteTras
    if metros < 0 Then
        metros:=0
    passos:=passos + 1
    return passos
end ChegarEmCasa

```

O procedimento irá calcular o tempo e os passos para que o bêbado chegue em casa, e de acordo com alguns teste que fiz, ele demorou em média 7 horas e 42 segundos, dando 181 passos.

Parte V: Estruturas Sequenciais – Pilhas, Filas e Listas

OBS: A pasta com todos os arquivos citados foram enviadas em conjunto na própria resposta à atividade. Mas caso aconteça algum imprevisto, tudo está em meu repositório no github:

<https://github.com/FerMatheus/Estrutura-de-Dados>

1. **Implemente uma classe Pilha vetorial considerando todas os seus métodos, para uma estrutura composta de índice inteiro e dois dados alfanuméricos.**
Solução:

Arquivo solução: Estrutura-de-Dados/Questao 1/Pilha.cs
Estrutura-de-Dados/Questao 1/T.cs

2. **[CALCULADORA POLONEZA] Implemente uma calculadora pós-fixada (notação Poloneza) usando pilhas. Considere as operações de soma, subtração, multiplicação e divisão com parênteses.**

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 2/CalculadoraPolonesa.cs

```

matheus@samsung-archlinux:~/Programming/CSharpProjects/Codes
[matheus@samsung-archlinux Codes]$ dotnet run
Escreva uma função no estilo polones:
(((5 2) - 4) + 7) /
1

```

3. [EMBARALHANDO] Considere que são fornecidas duas sequências de dados as quais serão empilhadas em duas pilhas distintas respectivamente. Transferir a sequência da pilha 2 para a pilha 1 e inverter a ordem de inclusão em seguida. Apresente o resultado da pilha 1 e da pilha 2 ao final desta operação.

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 3/Embaralhando.cs

```
matheus@samsung-archlinux:~/Programming/CSharpProjects/Codes
[matheus@samsung-archlinux:~/Programming/CSharpProjects/Codes]$ dotnet run
Pilhas antes de embaralhar:
Pilha 1:
5
4
3
2
1
Pilha 2:
10
9
8
7
6

Pilhas depois de embaralhar:
6
7
8
9
10

1
2
3
4
5
```

4. [CRIPTOGRAMA] Uma mensagem foi transferida e criptografada através de uma pilha com intermédio. A pilha com intermédio tem tamanho máximo k , enquanto a pilha sem intermédio tem tamanho n , onde $n \gg k$. Considere que à medida que os dados entram na pilha sem intermédio eles também são transferidos para a pilha intermédio até que a mesma seja preenchida, ela descarrega em uma saída, em seguida descarrega o conteúdo da pilha sem intermédio. Continuando a ler a mensagem. Supondo que a mensagem tem tamanho $m \gg n \gg k$, faça um algoritmo que decodifique esta mensagem. Considere a mensagem composta de strings de tamanho 1.

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 4/Criptograma.cs

5. Implemente uma classe Fila vetorial do tipo circular. Verifique o seu funcionamento e a sua corretude.

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 5/FilaCircular.cs

6. [FURANDO-FILA] Uma fila é implementada de um modo que permite a inclusão de um novo elemento que pode ganhar posições dependendo do seu conhecimento em relação aos que lá estão, conforme segue... “Num grande show popular Pedro resolveu comprar seu ingresso e conta com a possibilidade de encontrar algum conhecido, se houver algum conhecido Pedro passa na frente de todos os que vêm antes e segue para a posição do conhecido à frente da fila. Se demorar muito ele perde a fila porque todos têm um tempo de atendimento próprio, e ele leva k unidades de tempo a cada pessoa que está na sua frente até o seu conhecido. Implemente um processo que permita apresentar a entrada de Pedro na fila sabendo que ele pode ou não ter um conhecido, e que ele tem sempre que entrar na fila pelo fundo, e jamais pode pular conhecidos. A fila anda em função de um relógio, que opera para todos cujo atendimento depende da compra de ingressos de cada um.

7. Por que Pilhas e Filas têm que ser implementadas como vetores estáticos? Se não for, qual a alternativa e qual o preço que se paga por isto?

Solução:

Pilhas e Filas são estruturas de dados de rápida leitura e escrita, e por tanto, devem ser implementadas de modo estático a fim de preservar essa característica, pois teriam acesso a uma memória muito mais veloz e valiosa que é a memória de programa.

Ao serem implementadas de forma dinâmica, seria necessário abrir pequenos blocos na memória dinâmica (Heap), já tendo o custo de acessá-la. Ao necessitar alocar mais memória, mais custo de acesso e abertura na memória desse espaço, levaria a uma perda na otimização e maior custo computacional.

8. [SIMULADOR DE TRÁFEGO MULTIUSUÁRIO] Escreva um programa para simular um sistema de computador multiusuário, como segue: cada usuário tem um ID exclusivo e deseja executar uma operação no computador. Entretanto somente uma transação pode ser processada por vez. Cada linha de entrada representa um único usuário e contém o ID do usuário seguido de uma hora de início e a duração de sua transação. O programa deve simular o sistema e imprimir uma mensagem contendo o ID do usuário e a hora em que a transação começou e terminou. No final da simulação, ele deve imprimir o tempo médio geral de espera para uma transação (o tempo de espera de uma transação é o intervalo de tempo entre a hora em que a transação foi solicitada e a hora em que ela foi iniciada).

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 8/Usuario.cs
Estrutura-de-Dados/Questao 8/Ticket.cs
Estrutura-de-Dados/Questao 8/App.cs

9. [EMPILHANDO CAIXAS] Uma empilhadeira carrega caixas de 6, 10 e 14 toneladas. Há três pilhas: A, B e C. A pilha A é onde se encontram todas as caixas que chegam no depósito. Com um detalhe: caixas de maior peso não podem ser sobrepostas às de menor peso. Elabore um método que permite o controle das caixas, de forma que caso uma caixa de maior peso do que uma que já está em A deve ser empilhada, todas as caixas que estão em A são movidas para as pilhas auxiliares B (contendo somente caixas de 10t) e C (contendo somente caixas de 6t) até que se possa empilhar a nova caixa. Depois todas as caixas são movidas de volta para a pilha A.

Solução:

Arquivo solução: Estrutura-de-Dados/Questão 9/Empilhadeira.cs
Estrutura-de-Dados/Questão 9/Caixa.cs

Entrada:

```
Caixa[] caixas = { new(6), new(6), new(10), new(10), new(14), new(14) }; // entre parenteses temos os pesos
```

```
Pilha A:  
Caixa { Peso = 14 }  
Caixa { Peso = 14 }  
  
Pilha B:  
Caixa { Peso = 10 }  
Caixa { Peso = 10 }  
  
Pilha C:  
Caixa { Peso = 6 }  
Caixa { Peso = 6 }
```



```
Pilha A:  
Caixa { Peso = 6 }  
Caixa { Peso = 6 }  
Caixa { Peso = 10 }  
Caixa { Peso = 10 }  
Caixa { Peso = 14 }  
Caixa { Peso = 14 }  
  
Pilha B:  
Pilha Vazia!  
Pilha C:  
Pilha Vazia!
```

10. Implemente uma classe lista encadeada considerando os seguintes aspectos:

- (a) Permite a criação, inserção, busca, retirada e eliminação da lista como lista encadeada simples não ordenada;

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 10/ListaEncaeadaSimples.cs

- (b) Permite a criação, inserção, busca, retirada e eliminação da lista como lista encadeada simples e circular não ordenada ou ordenada;

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 10/ListaEncadeadaSimplesCircular.cs

- (c) Permite a criação, inserção, busca, retirada e eliminação da lista como lista encadeada simples não ordenada ou ordenada;

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 10/ListaEncadeadaSimplesOrdenadaTambem.cs

- (d) Permite a criação, inserção, busca, retirada e eliminação da lista como lista duplamente encadeada e circular não ordenada ou ordenada;

Solução:

Arquivo solução: Estrutura-de-Dados/Questao 10/ListaDuplamenteEncadeadaCircular.cs

- (e) Permite a criação, inserção, busca, retirada e eliminação da lista como lista de blocos não ordenada ou ordenada;

Solução:

Arquivo solução: Estrutura-de-Dados/ Questao 10/ListaBlocos.cs

- (f) Permite a criação, inserção, busca, retirada e eliminação da lista como vetor de listas duplamente encadeadas não ordenadas ou ordenadas;
(g) Permite a criação, inserção, busca, retirada e eliminação da lista como matriz de listas duplamente encadeadas não ordenadas ou ordenadas.

11. Verifique o funcionamento da alocação de ponteiros e acesso à memória de um conjunto de dados da estrutura abaixo que se organizam em uma lista encadeada simples, e responda:

- A. Apresente um gráfico de acessibilidade aos dados das estruturas considerando uma implementação feita em JAVA;
B. Faça o mesmo para a linguagem C;
C. Qual a conclusão que você tira sobre o acesso à memória pela indicação por ponteiro: Ela é constante ou linear? Como você justifica isto. (Sugestão: Use os resultados dos itens anteriores para justificar a sua resposta)

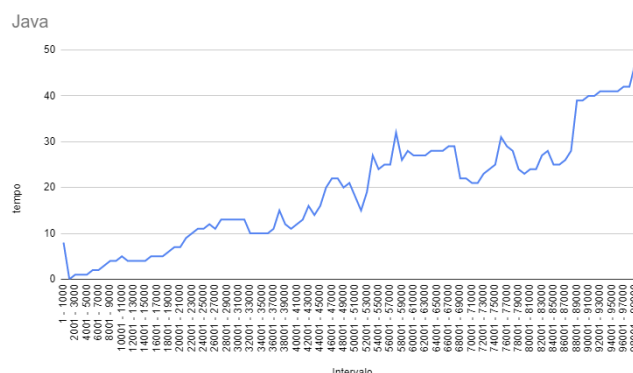
Estrutura de dados básica para o teste

L : Record
Nome : string[50];
Idade: word;
Endereco:string[100];
Número: word;
Complemento: string[15];
CEP: word;
Prox : ^L;
End;

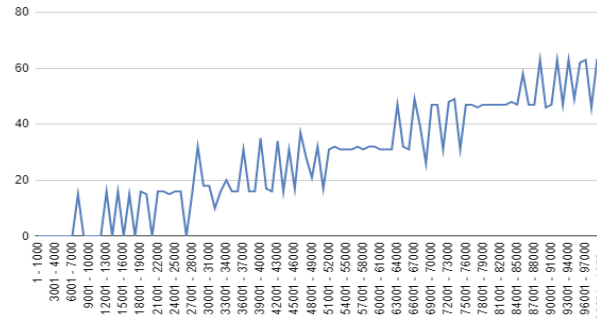
Fazer alocação aleatória de 100mil indivíduos L na lista.
Testar o acesso conforme a faixa de valores dos indivíduos na memória. Por exemplo, 100 buscas aleatórias nas faixas de 1 a 1000, 1001 a 2000, ..., 99001 a 200.000

Solução:

a) Java



b) C



c) Concluo que nas duas implementações tivemos muita variação, tanto no começo da lista, quanto no seu fim, quanto ao acesso à memória. Mas no geral, demonstram comportamento linear, pois a medida que se afastaram, o tempo de acesso também aumentou.

12. [REPRESENTAÇÃO INFINITA] Considerando que números irracionais são de grande utilidade para diversas aplicações, considere representar os números \sqrt{n} , onde n é primo, e os mais conhecidos: π (pi) e e (neperiano). Desenvolva uma classe que permita calcular com dada precisão estes números.

Solução:

Arquivo solução: *Estrutura-de-Dados/Questao 12/Pi.java*
Estrutura-de-Dados/Questao 12/RaizN.java
Estrutura-de-Dados/Questao 12/Euler.java

13. [MÁQUINA DE CALCULAR INFINITA] Tendo realizado a máquina de calcular em notação polonesa, agora permita fazer operações de inversão, logaritmo e exponenciais, além das básicas de somar, subtrair, multiplicar, dividir, percentual e mudança de sinal.

Solução:

Arquivo solução: *Estrutura-de-Dados/Questao 13/CalculadoraPolonesaAvançada.cs*

14. [PRODUÇÃO DE POTES] Suponha que uma fábrica produz potes de plástico com tampas coloridas (vermelhas, verdes e azuis). Um funcionário da linha de montagem monta 100 potes por dia. No início do dia ele recebe a ordem de serviço, que contém o número de potes a montar de cada cor. Outro funcionário monta kits com potes de três cores diferentes. Especifique as estruturas de dados para os tipos de potes e kits, e implemente uma aplicação que represente o processo de montagem dos kits.

Solução:

Arquivo solução: *Estrutura-de-Dados/Questao 14/Questao14.java*
Estrutura-de-Dados/Questao 12/Pilha.java

15. [PREPARANDO UM CALENDÁRIO DE PROVAS] Numa faculdade os alunos do curso de computação fazem disciplinas nos centros de Física, Matemática e Computação. Na Física existem 3 disciplinas, na Matemática existem 4 disciplinas e na Computação existem 8 disciplinas. Haverá prova na faculdade de todas as disciplinas e é necessário fazer um calendário de provas ao longo de uma semana com cinco dias úteis e um turno de provas sendo até duas provas de 2h por turno, e intervalo de 20min entre provas. Considerando que as salas de aula da universidade têm 40 lugares, e que há 401 alunos dos quais nenhum deles faz mais que cinco disciplinas. Utilizando suas implementações, prepare um conjunto de dados de alunos aleatoriamente que frequenta um conjunto de disciplinas em conformidade com os limites estabelecidos e prepare um calendário de provas que atenda aos alunos e à oferta de lugares nas salas de aula.

16. Considerando que se deseja usar o método radix-sort sobre um conjunto de dados em um arquivo texto no formato alfanumérico, cujo único campo tem tamanho máximo desconhecido, descreva um procedimento radix considerando que cada dado será convertido em formato hexadecimal equivalente em seguida resolvida a ordenação, por fim transformado-se de volta para o formato original do arquivo ordenado. Gere um exemplo com um arquivo texto incluindo 100 dados aleatórios.

Solução:

Arquivo solução: *Estrutura-de-Dados/Questao 16/RadixSort.cs*
Estrutura-de-Dados/Questao 16/ListRadix.cs
Estrutura-de-Dados/Questao 16/NoHexa.cs