

Estruturas de Dados II - 2012-2

L3

Algoritmos de Ordenação e Complexidade de Algoritmos

Nas questões abaixo, você pode usar as seguintes funções:

qsort(v, N): ordena o vetor v de tamanho N em $O(N \cdot \log N)$

vmax(v, N): retorna a posição do maior elemento de um vetor de tamanho N em $O(N)$

vmin(v, N): retorna a posição do menor elemento de um vetor de tamanho N em $O(N)$

bsearch(v, N, num): retorna a posição do elemento **num** em um vetor de N elementos em $O(\log N)$.

Se não encontrar o elemento **num** no vetor, retorna -1.

- 1) Dados dois vetores de inteiros de mesmo tamanho N e asseguradamente sem nenhum inteiro repetido em cada um, crie um algoritmo $O(N \cdot \log N)$ para informar quantos inteiros repetidos há em ambos os vetores. **Exemplo:** nos vetores {7, 1, 5, 3} e {2, 1, 4, 7} há dois inteiros repetidos em ambos, que são os inteiros 1 e 7.

- 2) Suponha que queira ordenar um vetor de cadeias de caracteres (*strings*) com, no máximo, 5 caracteres cada uma usando o algoritmo *radixsort*. Explique como modificaria o algoritmo para isso e de que forma essa ordenação seria $O(N)$. Dica: $O(5 \cdot N) \approx O(N)$.

- 3) Um aluno de computação criou o código abaixo e apresentou a seu professor dizendo que se tratava de um novo tipo de ordenação usando uma função recursiva. Seu professor rebateu antes mesmo de ver o código dizendo que há alguns algoritmos de ordenação recursivos e isso não é novidade. Ao ler o código, o professor identificou que ele corresponde a um algoritmo de ordenação conhecido. Qual é esse algoritmo e justifique por que se trata desse algoritmo.

```
function recursive_sort(v, N)
  for j=1,N do
    if v[j]>v[j+1] then
      v[j], v[j+1] = v[j+1], v[j]
    end
  end
  if N-1>1 then recursive_sort(v, N-1) end
end
```

- 4) Qual o melhor algoritmo de ordenação para usar nos casos a seguir e por quê?
- Ordenar um vetor com as notas de alunos em uma turma, cada nota representada por um inteiro variando de 0 a 100.
 - Ordenar um vetor qualquer com 4 elementos.
 - Ordenar um vetor qualquer com 4000 elementos.
 - Ordenar as medidas de altura de um vetor contendo as alturas de todos os prédios de São Luís representadas por valores inteiros em metros.

- 5) Informe a complexidade dos algoritmos a seguir.

- a) Busca binária em um vetor ordenado.

```
binSearch(v, ini, fim, num)
  if ini>fim then return -1 end
  middle = (ini+fim)/2
  if v[middle]==num then
    return middle
  elseif num>v[middle] then
    return
    binSearch(v, middle+1, fim, num)
  else
    return
    binSearch(v, ini, middle-1, num)
  end
end
```

end

b) Fatores primos de um inteiro X ($1 \leq X \leq 2^Y$), dado um vetor V com N primos de 2 a $2^{Y/2}$.

```
primefactors(X, V)
factors = {}
for i=1,#V do
  if X%V[i]==0 then factors[ V[i] ]=0 end
  while X%V[i]==0 do
    factors[ V[i] ] = factors[ V[i] ] + 1
    X = X/V[i]
  end -- while
end -- for
if not X==1 then factors[X] = 1 end
return factors
end
```