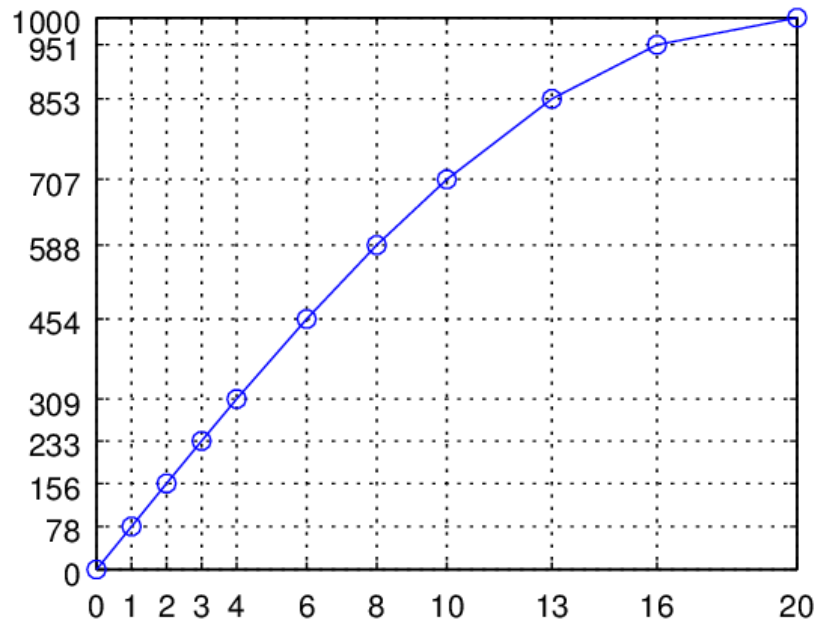


Aula prática nº 9 – Pesquisa e ordenação

Exercícios

- 1) Escreva um programa que leia um ficheiro de texto e, com base no mesmo, gere uma lista de todas as palavras existentes ordenadas lexicograficamente. Acrescente ao programa a opção de listar as palavras por ordem decrescente do número de ocorrências.
- 2) Escreva um programa que, dada a lista de alunos de FP com os seus Nomes, NMEC's, Turma, NotaAPi e NotaATP num ficheiro CSV (`fp.csv`), permita imprimir no terminal essa mesma lista por qualquer ordem (crescente/decrescente), e considerando qualquer um dos campos. Se pretender, implemente este programa com base num menu em que seja possível escolher o campo de ordenação pretendido. Tenha em atenção a escolha da estrutura de dados adequada para representar a informação contida no ficheiro.
- 3) Faça uma função que calcule a mediana de uma lista de valores. A mediana é um valor que é maior que metade dos valores da lista e menor que a outra metade. Se a lista tiver um número ímpar de valores, a mediana é o valor do meio da lista ordenada. Se a lista tiver um número par de valores, a mediana é a média dos valores do meio, depois de ordenados.
- 4) Nos sistemas Unix/Linux, o ficheiro `/usr/share/dict/words` contém uma lista de palavras de língua inglesa, por ordem. Leia essas palavras para uma lista e, usando uma função de pesquisa binária, descubra quantas palavras começam por “ab”, sem ter de percorrer tudo. Sugestão: procure a primeira palavra com “ab” e a primeira com “ac” e subtraia os índices. E quantas palavras começam por “tb”? Nenhuma? Então qual é primeira letra, maior que ‘b’, que ocorre após um “t”, nas palavras inglesas?
- 5) Usando o mesmo princípio, faça uma função que indique todas as letras que podem suceder a um certo prefixo. Pode usar esta função num sistema de escrita inteligente, que vai apresentando as letras possíveis para completar um certo prefixo já introduzido. Quando o utilizador introduz mais uma letra, é acrescentada ao prefixo anterior e apresenta-se nova lista de possibilidades...

- 6) Uma técnica muito usada em computação para evitar cálculos demorados de certas funções complexas consiste em fazer uma pesquisa numa tabela de valores pré-calculados, e depois fazer interpolação entre os valores mais próximos. (A este tipo de tabela chama-se look up table, ou LUT). Por exemplo a função seguinte está definida através de uma tabela de 11 pontos (x, y) .



```

xk = [0, 1, 2, 3, 4, 6, 8, 10, 13, 16, 20]
yk = [0, 78, 156, 233, 309, 454, 588, 707, 853, 951, 1000]

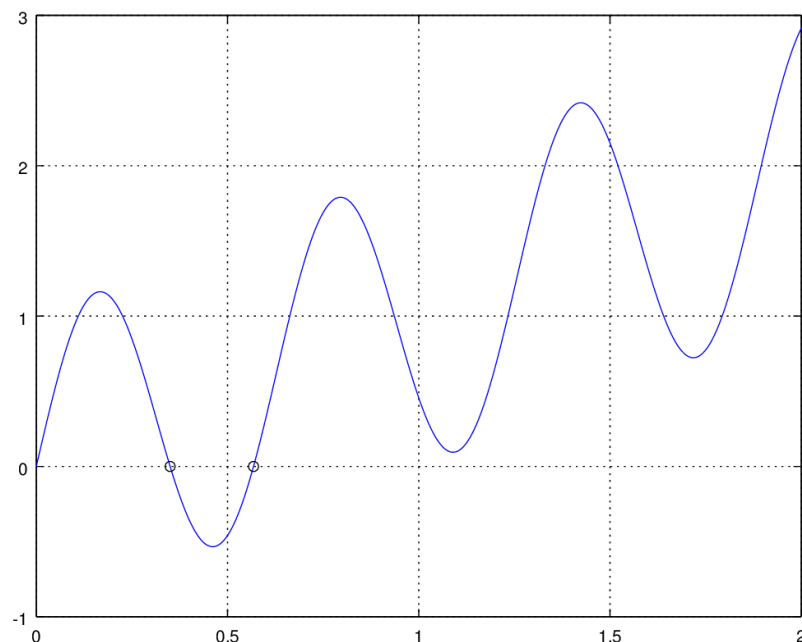
```

Para calcular $y = f(x)$, com $x=9$ por exemplo, teria que descobrir que $x_k[6] \leq x < x_k[7]$, e depois fazer interpolação linear entre os pontos $(x_k[6], y_k[6])$ e $(x_k[7], y_k[7])$, o que se pode fazer pela expressão:

$$y = y_k[7] - y_k[6] \frac{x_k[7] - x_k[6]}{x_k[7] - x_k[6]} (x - x_k[6]) + y_k[6] = 647.5$$

Faça uma função que, dadas a tabela x_k (já ordenada), a tabela y_k e o valor x , calcula o resultado correspondente y . Como x_k está ordenada, pode usar pesquisa binária para encontrar o intervalo que contém x .

- 7) A função $y(x) = x + \sin(10x)$ (ver gráfico abaixo) tem pelo menos uma raiz (um zero) no intervalo $[0.25, 0.5]$. Sabemos isso porque é uma função contínua e muda de sinal entre os extremos desse intervalo: $y(0.25) > 0 \wedge y(0.5) < 0$. De facto, a função tem uma outra raiz no intervalo $[0.5, 1]$, porque $y(0.5) < 0 \wedge y(1) > 0$. Encontre cada uma dessas raízes usando o método da bissecção. Trata-se de um método de aproximações sucessivas que, em cada iteração, vai reduzindo o intervalo que contém a raiz para metade. Quando a amplitude do intervalo for inferior à precisão pretendida, paramos. Isto é uma forma de pesquisa binária, mas em vez de pesquisar uma lista, tem de “pesquisar” uma função de variável real. (Repare que em Python pode passar uma função como argumento para outra função.)



- 8) Implemente o algoritmo de ordenação por inserção.
- 9) Implemente o algoritmo de pesquisa binária (nas suas duas variantes) utilizando recursividade.