



Pré-visualizar teste: Exame

Informações do teste

Descrição **Prova individual sem consulta.** Qualquer evidência do contrário nas respostas dadas implicará a reprovação à disciplina e demais consequências previstas no regulamento da Universidade.

Instruções

Teste Este teste tem um limite de tempo de 2 horas e 30 minutos. Este teste será salvo e enviado automaticamente quando o prazo expirar.
cronometrado Avisos aparecem quando restar **metade do tempo, 5 minutos, 1 minuto e 30 segundos.** [O temporizador não aparece ao fazer uma visualização prévia deste Teste]

Várias Não permitido. Este teste só pode ser feito uma vez.

Estado de Conclusão da Pergunta:

conclusao timer continuara a contar se voce sair do teste.

PERGUNTA 1

2 pontos**Salvar resposta**

Considere o problema de, num array de inteiros, determinar o comprimento do maior subarray sem elementos repetidos.

Para isso definiu-se uma função auxiliar que calcula o tamanho do maior sufixo sem elementos repetidos.

```
int auxiliar (int v[], int N) {
    int i, j, found=0;
    // Pre: N>0
    i=N-2; found = 0;
    while (i>=0 && !found) {
        // Inv: ???
        j=N-1;
        while (v[j] != v[i]) j--;
        if (j==i) i--; else found = 1;
    }
    // Pós: -1<=i<=N-2 && nrep (v,i+1,N) && (i<0 || exists (i<k<N) v[k] == v[i])
    return (N-i-1);
}
```

Clique em Salvar e Enviar para salvar e enviar. Clique em Salvar todas as respostas para salvar todas as respostas

```
        if (t>r) r = t;  
    return r;  
}
```

Apresente um invariante para o ciclo exterior da função `auxiliar` que permita verificar a respectiva pós-condição. O predicado `nrep(v, i, j)` testa que não existem repetidos no array `v` entre as posições `i` e `j-1`.

				Parágrafo	▼	Arial	▼	3 (12pt)	▼										

Caminho: 0 Words: 0

▼ Estado de Conclusão da Pergunta:

PERGUNTA 2

2 pontos[Salvar resposta](#)

De forma a analisar a complexidade da função `maiorsubseq`, identifique o melhor e pior casos da execução da função `auxiliar` definida acima, considerando o número de comparações efectuadas entre elementos do array. Para o pior caso diga também, justificando, qual o número de comparações efectuadas.

				Parágrafo	▼	Arial	▼	3 (12pt)	▼										

Caminho: p Words: 0

PERGUNTA 3

2 pontos[Salvar resposta](#)

O pior caso da execução de `maiorsubseq` corresponde ao pior caso da execução da função `auxiliar`. Para este caso diga, justificando, qual o número de comparações efectuadas pela função `maiorsubseq` entre elementos do array. Para isso, apresente e resolva uma recorrência que traduza esse número em função do tamanho do array.

Clique em Salvar e Enviar para salvar e enviar. Clique em Salvar todas as respostas para salvar todas as respostas

Caminho: p

Words: 0

PERGUNTA 4**2 pontos****Salvar resposta**

Tendo em consideração que o pior caso da execução de `maiorsubseq` corresponde ao pior caso da execução da função auxiliar, apresente uma definição otimizada da função `maiorsubseq` que tira partido desse facto, e em que o tempo de execução nesse caso é substancialmente reduzido. Identifique ainda, justificando, o pior caso de execução dessa versão otimizada.

Parágrafo ▼ Arial ▼ 3 (12pt) ▼

▼ Estado de Conclusão da Pergunta:

Mashups

Caminho: p

Words: 0

PERGUNTA 5**2 pontos****Salvar resposta**

Uma forma de otimizar este algoritmo passa por calcular, para cada posição k do array, a última posição (entre 0 e $k-1$) onde esse elemento já apareceu no array (-1 caso seja a primeira ocorrência). A seguinte função faz esse cálculo de uma forma muito pouco eficiente ($O(N^2)$).

```
void ultimo (int v[], int N, int ult[]) {  
    int i,j;  
    for (i=0;i<N;i++)  
        for (j=i-1; j>=0 && v[i] != v[j]; j--);  
    ult[i] = j;  
}
```

Uma forma alternativa de preencher o array `ult` consiste em usar uma tabela de Hash onde se percorre o array `v` da esquerda para a direita e se guarda, para cada elemento a última posição onde ele correu. Mostre como poderia declarar a tabela de Hash que seria necessária para implementar esta versão otimizada da função `ultimo`, por forma a garantir que o tempo

Clique em Salvar e Enviar para salvar e enviar. Clique em Salvar todas as respostas para salvar todas as respostas



Caminho: p

Words: 0

PERGUNTA 6**2 pontos****Salvar resposta**

Escreva uma pós-condição para a função `ultimo`. Tente que esta seja o mais informativa possível.

				Parágrafo	▼	Arial	▼	3 (12pt)	▼										

▼ Estado de Conclusão da Pergunta:

Caminho: p

Words: 0

PERGUNTA 7**2 pontos****Salvar resposta**

Assumindo que, num array aleatório, a probabilidade de dois elementos serem iguais é dada por p , apresente uma expressão que calcule o número médio de comparações entre elementos do array efectuadas pela função `ultimo` apresentada acima.

				Parágrafo	▼	Arial	▼	3 (12pt)	▼										

Caminho: p

Words: 0

Clique em Salvar e Enviar para salvar e enviar. Clique em Salvar todas as respostas para salvar todas as respostas


ant contem uma floresta de arvores de travessia.

```
int travessiaBFTreeComplete(Graph g, int ant[]) {
    int i;
    for(i=0; i<NV; i++) ant[i] = -2;
    for(i=0; i<NV; i++) if (ant[i] == -2) travessiaBFTree(g,i,ant);
}
```

Considere agora uma execução desta função sobre um grafo não orientado com 6 vértices, que produziu o seguinte vector ant [-1,-1,0,4,5,0].

Indique se é possível usando este vector responder a cada uma das seguintes questões, e em caso afirmativo diga qual é a resposta respectiva:

1. Os vértices 2 e 3 pertencem ao mesmo componente ligado do grafo?
2. Qual é a distância (comprimento do caminho mais curto) entre os vértices 2 e 3?
3. Qual é a distância entre os vértices 4 e 0?

				Parágrafo ▼	Arial ▼	3 (12pt) ▼												
																		

▼ Estado de Conclusão da Pergunta:

PERGUNTA 9

2 pontos

Salvar resposta

Complete a definição da seguinte função, que "pinta" os vértices de um grafo não orientado com o componente a que pertencem, i.e., preenche o vector comp colocando em cada posição u um número inteiro (0, 1, 2, ...) que identifica o componente a que pertence o vértice u. A função começa por chamar a função travessiaBFTreeComplete, trabalhando depois a partir do vector ant (note que dois vértices pertencem ao mesmo componente se fizerem parte da mesma árvore de travessia).

```
int componentes (Graph g, int comp[]) {
    int ant[NV];
    travessiaBFTreeComplete(g, ant);
    // Completar
}
```

				Parágrafo ▼	Arial ▼	3 (12pt) ▼												
																		

Caminho: n

Words: 0

Clique em Salvar e Enviar para salvar e enviar. Clique em Salvar todas as respostas para salvar todas as respostas

↘ Estado de Conclusão da Pergunta:

https://elearning.uminho.pt/webapps/assessment/take/launch.jsp?course_assessment_id=_19506_1&course_id=_45159_1&content_id=_1097539_1&step=null 6/6