Algoritmos e Complexidade — Universidade do Minho
Teste 29 de Outubro de 2024

1. Especificação (2 valores)

Considere a seguinte função que, dado um número inteiro x não negativo calcula a sua raiz quadrada inteira (i.e., o maior número inteiro `r` tal que `r*r <= x`). Por exemplo, `intSqrt (25) = 5` e `intSqrt (35) = 5`.
Complete a especificação da função fornecendo uma pré e pós condições apropriadas.

```
int intSqrt (int x) {
     // Pre: …
     int r=0, p=1, i=1;
     while (p<=x){
          r+=1; i+=2; p+=i;
     }
     // Pos: r*r <= x && …
     return r ;
}
```

2. Correção (6 valores)

Escreva um invariante e um variante que permitam provar a **correcção total** da função da Questão 1.

3. Complexidade de algoritmos recursivos (4 valores)

A função `int subsetsum (int x, int v[], int N)` definida abaixo testa se existe um subconjunto dos elementos do array `v` cuja soma seja `x`.
Apresente e resolva uma recorrência que traduza o número de acessos ao array no pior caso (comece por identificar esse caso).

```
int subsetsum (int x, int v[], int N){
    if (x==0) return 1;
    if (N==0) return 0;
    return (subsetsum (x,      v+1, N-1) ||
```

```
            subsetsum (x-v[0], v+1, N-1));
}.
```

4. Complexidade algoritmos iterativos (8 valores)

A função `inc` recebe como argumento uma string de dígitos (que representa um número) e adiciona-lhe `1`, devolvendo `1` se ocorreu um overflow ou `0` noutro caso. Por exemplo, se `s="23499"` o resultado de invocarmos `inc(s,5)` é `0` e `s` passa a conter a string `"23500"`.

```
int inc (char s[], int N){
    int i=N-1;
    while (i>=0 && s[i] =='9')
        s[i--] = '0';
    if (i<0) return 1;
    s[i] = s[i] + 1;
    return 0;
}
```

1. Considerando o custo desta função como o número de caracteres de `s` que são alterados, identifique o pior caso da execução da função e, para esse caso, indique qual o seu custo.
2. Calcule ainda o custo médio desta função.
3. Usando a função de potencial ϕ (`s`) = (soma dos dígitos de `s`) / `9`, calcule o custo amortizado da função `inc`.