UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
ESCOLA DE INFORMÁTICA APLICADA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Quarto Trabalho
Programação Dinâmica - Segmentos de Soma Máxima

Bruno Lírio Alves
Pedro Paulo Gouveia
Thales Veras

Vânia Felix

Rio de Janeiro, 12 de novembro de 2013.

2. Motivação
2.1 Definição do Problema

Dada uma grandeza escalar que evolui com o tempo, aumentando ou diminuindo uma vez por unidade de tempo u, de maneira irregular e dado o registro das variações desta grandeza ao longo de $t * u$ com $t \geq 1$, queremos encontrar um intervalo de tempo em que a variação acumulada tenha sido máxima.

-Entrada: Um conjunto numérico, por conveniência, vamos supor que este seja um conjunto de inteiros.
-Questão: Como encontrar a maior subsequência no conjunto.
-Saída:A soma da maior sequência contínua.

2.2. Exemplos de Instâncias do Problema

Dada a sequência abaixo qual é a maior subsequênca?

20	-30	15	-10	30	-20	-30	30

3. Algoritmo
3.1. Descrição da ideia do algoritmo

Dado um vetor A com índices variando de p até r, ou seja A[p...r], como dito, queremos encontrar a maior subsequência em A, um algoritmo de força bruta calcularia todas as combinações de somas entre subsegmentos, variando tanto o comprimento como o início de cada subsegmento.

Pela abordagem da programação dinâmica, vamos resolver um subproblema relacionado primeiramente, ou seja, vamos calcular as somas de subsegmentos mas, evitando o recálculo de valores através do armazenamento destes.

Para isso, vamos definir a firmeza:
firmeza é a maior soma em A[i] + … + A[r] com $p \leq i \leq r$.
Assim, o segmento de soma máxima é a maior das firmezas entre A[p...r], A[p...r-1], A[p...r-2], etc.

A firmeza do vetor tem uma propriedade recursiva que a segmento de soma máxima não tem. Suponha que A[i] + … + A[r] é a firmeza de A[p...r]. Se $i \leq r-1$, A[i] + … + A[r-1] é a firmeza de A[p...r-1]. (De fato, se A[p...r-1] tivesse firmeza maior então existiria $h \leq r-1$ tal que A[h] +... + A[r-1] > A[i] + … + A[r], o que é impossível).

Se denotarmos a firmeza de A[p...q] por F[q], podemos resumir a propriedade recursiva por meio de uma recorrência: para qualquer vetor A[p...r],

F[r] = max(F[r-1] + A[r], A[r])

Em outras palavras, F[r] = F[r-1] + A[r] a menos que F[r-1] seja negativo, caso em que F[r] = A[r].

A recorrência serve de base para a ideia do algoritmo, que calcula a solidez de A[p...r] supondo $p \leq r$.

De maneira mais informal podemos dizer que firmeza é então a soma dos elementos de forma sequencial, do vetor original, na qual para cada índice do vetor original temos uma firmeza correspondente e a cada soma negativa a firmeza é “resetada” e a ela atribuído o valor do elemento de índice seguinte do vetor original.

O algoritmo então percorre o vetor original e a cada A[q] é verificado se a soma anterior (F[q-1]) é positiva, se for, adiciona-se A[q] a F[q] (próximo), se não for a F[q] recomeça a partir de A[q+1]. Ao final a maior firmeza é retornada.

3.2. A partir da ideia, mostrar exemplos* de solução para as instâncias apresentadas

20	-30	15	-10	30	-20	-30	30

20	-10	15	5	35	15	-15	30

3.3. Descrição Formal do Algoritmo

```
SomaMaxima(A, p, r)
F[p] ← A[p]
para q ← p + 1 até r faça
        se F[q-1] > 0
                então F[q] ← F[q-1] + A[q]
                senão F[q] ← A[q]
x ← F[p]
para q ← p+1 até r faça
        se F[q] > x então x ← F[q]
devolva x
```

3.4. Etapas da aplicação da técnica

1ª passo:

F[1] = $\boxed{20}$

2ª passo:

F[2] = $\boxed{20} + \boxed{-30} = \boxed{-10}$

3ª passo:

F[3] = $\boxed{15}$ (a soma anterior foi descartada por ser menor que 0 e a nova firmeza foi estabelecida como sendo o elemento seguinte do vetor original)

4ºpasso:

F[4] = $\boxed{15} + \boxed{-10} = \boxed{5}$

5º passo:

F[5] = $\boxed{5} + \boxed{30} = \boxed{35}$

6º passo:

F[6] = $\boxed{35} + \boxed{-20} = \boxed{15}$

7ºpasso:

F[7] = $\boxed{15} + \boxed{-30} = \boxed{-15}$

8ºpasso:

F[8] = $\boxed{30}$ (mais uma vez o valor anterior é descartado por ser negativo e o elemento do vetor original é estabelecido como sendo a firmeza)

4. Análise do Algoritmo

4.1. Prova de corretude (mostrar que o algoritmo funciona de fato)

```
SomaMaxima(A, p, r)
F[p] ← A[p]
para q ← p + 1 até r faça
        se F[q-1] > 0
                então F[q] ← F[q-1] + A[q]
                senão F[q] ← A[q]
x ← F[p]
para q ← p+1 até r faça
        se F[q] > x então x ← F[q]
devolva x
```

Se observarmos a linha 2 e fizermos uso da propriedade recursiva da firmeza mencionada anteriormente, concluiremos que imediata antes que q seja comparado com r, F[q-1] é a firmeza de A[p…q-1], de maneira geral, F[j] é a firmeza de A[p...j].

Por fim, as linhas 7 e 8 escolhem a maior dentre as firmezas obtidas e assim temos o segmento de soma máxima.

4.2. Demonstração da complexidade/tempo do algoritmo (geralmente aborda-se o pior caso)

O algoritmo consome o tempo proporcional ao número de elementos n := r -p + 1 do vetor. O consumo de tempo do algoritmo está em $\Theta(n)$ e o algoritmo é linear.

5. Conclusão e Discussões

A programação dinâmica é a técnica mais eficiente para a resolução do problema, no entanto, se a entrada for de ordem colossal, pode chegar a acontecer estouro de recursos computacionais relacionados a memória.

5.1. Discutir as vantagens e desvantagens do procedimento adotado.

Vantagens:

- É o algoritmo mais eficiente
- Menor gasto de processamento
- Pode ser utilizada num grande número de problemas de otimização discreta.

Desvantagens:

- Algoritmo menos natural
- A complexidade espacial pode ser exponencial

5.2. É possível ter um algoritmo melhor?

Não, esse é o algoritmo ótimo para o problema dado.

6. Referências Bibliográficas

Segmento de Soma Máxima (2004). IME-USP. ***Site.*** Disponível em: <http://www.ime.usp.br/~cris/aulas/11_1_338/slides/>. Acesso em: 11 nov. 2013

Programação Dinâmica (2013). WIKIPEDIA. ***Site.*** Disponível em: <http://pt.wikipedia.org/wiki/Programação_Dinâmica>. Acesso em: 11 nov. 2013

Cormen, Algoritmos Teoria e Prática 2ª ed. ELSEVIER.