Programação Dinâmica
Segmento de Soma Máxima

Grupo 02:
Bruno Lírio
Thales Veras
Pedro Paulo

Motivação

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 maior soma de uma sequência contínua.

Exemplos Instâncias do Problema

Analisar a sequência abaixo qual é a maior subsequênca...

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

Descrição da ideia do Algoritmo

Um algoritmo força bruta, calcularia todas as combinações de somas dos subsegmentos, variando tanto o comprimento quanto 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 $A[i] + \ldots + A[r]$ 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 o segmento de soma máxima não tem. Suponha que $A[i] + \ldots + A[r]$ é a firmeza de A[p...r]. Se $i \leq r-1$, $A[i] + \ldots + A[r-1]$ 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 $A[h] + \ldots + A[r-1] > A[i] + \ldots + A[r]$, o que é impossível).

Descrição da ideia do algoritmo

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.

Exemplo de Solução para as instâncias

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

20	-10	15	5	35	15	-15	30

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
```

Etapas da Aplicação daTécnica

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

1º passo:

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

Etapas da Aplicação daTécnica

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

2º passo:

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

Etapas da Aplicação daTécnica

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

3º 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 A[q] do vetor original)

Etapas da Aplicação daTécnica

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

4º passo:

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

Etapas da Aplicação daTécnica

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

5º passo:

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

Etapas da Aplicação daTécnica

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

6º passo:

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

Etapas da Aplicação daTécnica

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

7º passo:

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

Etapas da Aplicação daTécnica

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

8º passo:

F[8] =	30

(mais uma vez o valor anterior é descartado por ser negativo e o elemento seguinte do vetor original é estabelecido como sendo a firmeza)

Etapas da Aplicação daTécnica

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

8º passo:

F[8] =

30

(mais uma vez o valor anterior é descartado por ser negativo e o elemento do vetor original é estabelecido como sendo a firmeza)

SOLUÇÃO:

20	-10	15	5	35	15	-15	30

Corretude

```
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 imediatamente 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]. Se variarmos j de p a r temos o conjunto das firmezas.

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

Complexidade

```
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
```

Pelos loops nas linhas 2 e 7, podemos ver que o algoritmo executa em tempo linear $\Theta(n)$, onde n é proporcional ao número de elementos do vetor, ou pelos índices dos loops $n = r - p + 1$.

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.

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

Possível ter um algoritmo melhor?

Não, o algoritmo é ótimo para o problema em questão.

Refêrencias Bibliograficas

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.