**MỤC LỤC**

[1. Nêu bài toán 3](#_Toc502682350)

[2. Mô tả chi tiết thuật toán 3](#_Toc502682351)

[2.1. Thuật toán 3](#_Toc502682352)

[2.2. Tính chính xác của thuật toán 3](#_Toc502682353)

[2.3. Giả mã 5](#_Toc502682354)

[3. Đánh giá độ phức tạp thuật toán 5](#_Toc502682355)

[4. Ví dụ 5](#_Toc502682356)

[5. Chương trình 6](#_Toc502682357)

1. **Nêu bài toán**

Bạn có một danh sách các hoạt động (s1, e1), (s2, e2), ..., (sn, en) được biểu thị bằng thời gian bắt đầu và kết thúc. Tất cả hoạt động đều hấp dẫn không kém đối với bạn. Chọn số hoạt động tối đa có thể được thực hiện bởi một người, giả định rằng một người chỉ có thể thực hiện một hoạt động tại một thời điểm.

1. **Mô tả chi tiết thuật toán**
   1. **Thuật toán**

* Dữ liệu vào:

+ Với mỗi hoạt động i có thời gian bắt đầu si và thời gian kết thúc ei.

* Dữ liệu ra

+ Số hoạt động tối đa có thể thực hiện bởi 1 người giả định rằng một người chỉ có thể thực hiện một hoạt động tại một thời điểm.

**Thuật toán Activity Selection dựa trên ý tưởng của giải thuật tham lam:**

Lựa chọn tham lam là luôn chọn hoạt động tiếp theo mà thời gian hoàn thành ít nhất trong số các hoạt động còn lại và thời gian bắt đầu lớn hơn hoặc bằng thời gian kết thúc của hoạt động đã chọn trước đó.

Các bước thực hiện của thuật toán:

1) Sắp xếp các hoạt động theo thời gian kết thúc của chúng.   
2) Khởi tạo Curent\_time=0;

3) Chọn hoạt động tiếp theo có thời gian bắt đầu lớn hơn hoặc bằng thời gian kết thúc của hoạt động đã chọn trước đó.

4) Quay lại bước 3 cho đến khi tất cả các hoạt động được kiểm tra.

* 1. **Tính chính xác của thuật toán**

Chúng ta cần chứng minh hai tính chất về thuật toán để đảm bảo tính chính xác.

    (1) **Tính hợp pháp**. Thuật toán tìm thấy lịch trình pháp lý của các hoạt động (nghĩa là nó không "lên lịch các hoạt động xung đột").

    (2) **Tối ưu.** Thuật toán tìm ra một lịch biểu hoạt động tối ưu (nghĩa là không có lịch biểu tốt hơn).

* Tính hợp pháp

Ở bước ba của thuật toán chúng ta lựa chọn hoạt động tiếp theo có thời gian

bắt đầu lớn hơn hoặc bằng thời gian kết thúc của hoạt động đã chọn trước đó đảm bảo tính tương thích của 2 hoạt động do đó các hoạt động không xung đột lẫn nhau.

* Tính tối ưu

Gọi S là lịch trình được tạo ra bởi thuật toán Activity Selection và S \* là lịch trình tối ưu nào đó.

Gọi | S | và | S \* | biểu thị số hoạt động ở S và S \* tương ứng.

Ta có: | S | ≤ | S \* | (do S\* là lịch trình tối ưu).

Cho f (i, S) biểu thị thời gian mà hoạt động thứ i kết thúc theo tiến độ S.

* Sử dụng phương pháp quy nạp chứng minh bổ đề:

Với bất kỳ 1 ≤ i ≤ | S |, chúng ta có f (i, S) ≤ f (i, S \*).

-TH i=1. Hoạt động đầu tiên mà thuật toán tham lam chọn phải là một hoạt động kết thúc không muộn hơn bất kỳ hoạt động khác, vì vậy f (1, S) ≤ f (1, S \*).

- Giả sử mệnh đề đúng với i=k tức là f(k, S) ≤ f (k, S \*). (1)

- Ta cần chứng minh mệnh đề đúng với i=k+1.

Từ (1) ta thấy hoạt động thứ i trong S kết thúc trước khi hoạt động thứ i trong S \* kết thúc. Thuật toán Activity Selection sẽ chọn 1 hoạt động (k+1) nào đó mà f(k+1,S) là nhỏ nhất trong tập hoạt động còn lại sau bước k. Do vậy kế hoạch thứ k+1 của S chắc chắn nằm trong đoạn (f(k,S),f(k+1,S\*)].

Do vậy f(k+1,S) ≤ f(k+1,S\*). Bổ đề được chứng minh.

* Dựa vào bổ đề chứng minh tính tối ưu của thuật toán.

Để chứng minh tính tối ưu của thuật toán cần chứng minh: |S|=|S\*|

Vì S\* là tối ưu do đó: | S | ≤ | S \* |

Giả sử | S | <| S \* |. Cho k = | S |.

Theo bổ đề thì f (k, S) ≤f (k, S \*), do đó hoạt động k trong S kết thúc không trễ hơn hoạt động k trong S \*.

Kể từ | S | <| S \* |, có một hoạt động (k + 1) trong S \*, và thời gian bắt đầu của nó

phải sau f (k, S \*) và do đó sau f (k, S). Như vậy sau khi thuật toán tham lam thêm hoạt động k của nó vào S, hoạt động (k + 1) từ S \* vẫn thuộc U.

    Nhưng thuật toán tham lam đã kết thúc sau những hoạt động k.

    Do vậy có mâu thuẫn, vì vậy giả định đã sai và | S \* | = | S |,

    Do đó, các thuật toán là tối ưu.

1. **Đánh giá độ phức tạp thuật toán**
2. **Ví dụ**

* Bộ dữ liệu 1

Xem xét 8 hoạt động sau đây với thời gian bắt đầu và kết thúc.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động(index)** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **Bắt đầu (start\_time)** | 1 | 0 | 1 | 4 | 2 | 5 | 3 | 4 |
| **Kết thúc (end\_time)** | 3 | 4 | 2 | 6 | 9 | 8 | 5 | 5 |

**Bước 1:** Sắp xếp các hoạt động theo thời gian kết thúc theo thứ tự tăng dần

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động(index)** | 3 | 1 | 2 | 7 | 8 | 4 | 6 | 5 |
| **Bắt đầu (start\_time)** | 1 | 1 | 0 | 3 | 4 | 4 | 5 | 2 |
| **Kết thúc (end\_time)** | 2 | 3 | 4 | 5 | 5 | 6 | 8 | 9 |

**Bước 2**: current\_time=0, S={};

**Bước 3**: Chọn hoạt động tiếp theo có thời gian bắt đầu lớn hơn hoặc bằng thời gian kết thúc của hoạt động đã chọn trước đó

S={3}, f(1,S)=current\_time=2;

S={3,7}, f(2,S)=current\_time=5;

S={3,7,6}, f(3,S)=current\_time=8;

* Bộ dữ liệu 2
* Xem xét 10 hoạt động sau đây với thời gian bắt đầu và kết thúc.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động(index)** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Bắt đầu (start\_time)** | 1 | 5 | 4 | 2 | 1 | 2 | 2 | 3 | 1 | 9 |
| **Kết thúc (end\_time)** | 50 | 8 | 6 | 8 | 7 | 6 | 4 | 8 | 2 | 10 |

**Bước 1:** Sắp xếp các hoạt động theo thời gian kết thúc theo thứ tự tăng dần

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động(index)** | 9 | 7 | 6 | 3 | 5 | 4 | 8 | 2 | 10 | 1 |
| **Bắt đầu (start\_time)** | 1 | 2 | 2 | 4 | 1 | 2 | 3 | 5 | 9 | 1 |
| **Kết thúc (end\_time)** | 2 | 4 | 6 | 6 | 7 | 8 | 8 | 8 | 10 | 50 |

**Bước 2**: current\_time=0, S={};

**Bước 3**: Chọn hoạt động tiếp theo có thời gian bắt đầu lớn hơn hoặc bằng thời gian kết thúc của hoạt động đã chọn trước đó.

S={9}, f(1,S)=current\_time=2;

S={9,7}, f(2,S)=current\_time=4;

S={9,7,3}, f(3,S)=current\_time=6;

S={9,7,3,10}, f(4,S)=current\_time=10

1. **Chương trình**

* **Mã nguồn**