



BÀI 4

HÀNG ĐỢI



TÌNH HUỐNG DẪN NHẬP

- Chúng ta hãy tưởng một hàng người đang chờ tính tiền ở một siêu thị;
- Một dãy các máy bay đang chờ hạ cách xuống một sân bay;
- Một dãy các công việc trong một hệ máy tính đang chờ một thiết bị ra nào đó, chẳng hạn như máy in;
- Trong các ví dụ trên, chúng ta nhận thấy chúng có một điểm chung đó là các mục trong mỗi hệ thống đó được phục vụ theo thứ tự như khi chúng đến; nghĩa là mục đầu tiên trong hàng sẽ được phục vụ đầu tiên.

Vậy hàng đợi được thể hiện như thế nào trong máy tính?



MỤC TIÊU

- Mô tả khái niệm về hàng đợi và phân biệt được hàng đợi với ngăn xếp;
- Đặc tả về hàng đợi;
- Trình bày chính xác các cách cài đặt hàng đợi bằng mảng và bằng danh sách liên kết được thực hiện bằng ngôn ngữ lập trình cụ thể;
- Sử dụng cấu trúc dữ liệu dạng hàng đợi để giải các bài toán.



NỘI DUNG



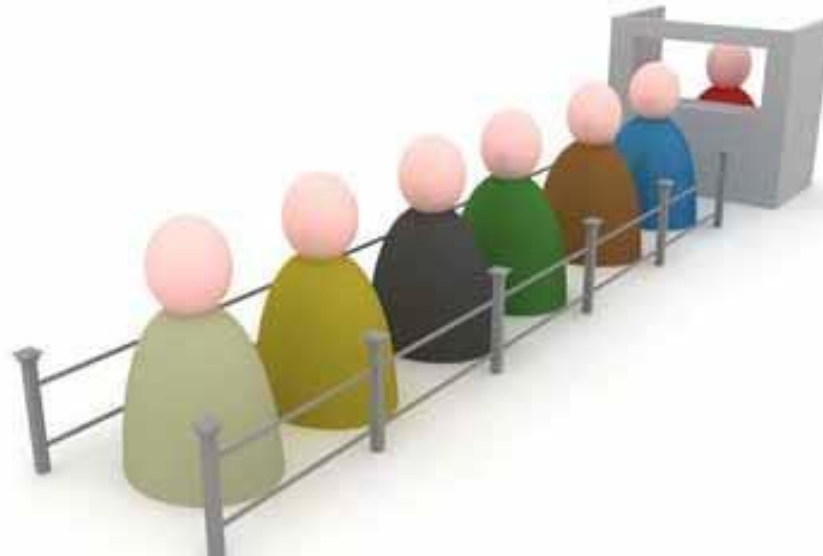
- Khái niệm về hàng đợi;
- Đặc tả hàng đợi;
- Các phương án cài đặt hàng đợi;
- Hàng đợi vòng tròn;
- Ứng dụng của hàng đợi.



1. KHÁI NIỆM VỀ HÀNG ĐỢI



1. KHÁI NIỆM VỀ HÀNG ĐỢI



Kiểu dữ liệu hàng đợi là kiểu dữ liệu trong đó phần tử được đưa vào trước sẽ được loại bỏ trước. Kiểu dữ liệu hàng đợi còn được gọi là kiểu dữ liệu FIFO (First-In First-Out), có nghĩa là vào trước ra trước.



2. ĐẶC TẢ HÀNG ĐỢI



2. ĐẶC TẢ HÀNG ĐỢI

Thao tác cơ bản đối với kiểu dữ liệu:

- Thêm một phần tử vào hàng đợi (enqueue);

Thao tác



Kết quả



Thêm phần tử 1



Hàng đợi



Như vậy sau 4 thao tác thêm phần tử, hàng đợi Q hiện thời có chứa phần tử 1,3,5,7.



2. ĐẶC TẢ HÀNG ĐỢI

Thao tác cơ bản đối với kiểu dữ liệu:

- Lấy một phần tử ra khỏi hàng đợi (enqueue).

Thao tác



Kết quả



Thao tác



Kết quả





3. CÁC PHƯƠNG ÁN CÀI ĐẶT HÀNG ĐỢI



3. CÁC PHƯƠNG ÁN CÀI ĐẶT HÀNG ĐỢI

- Trên thực tế có nhiều phương án cài đặt hàng đợi để đạt được tính hiệu quả về bộ nhớ và tốc độ xử lý;
- Trong khuôn khổ giáo trình này, chúng tôi chỉ giới thiệu hai phương pháp:
 - Cài đặt hàng đợi bằng mảng;
 - Cài đặt hàng đợi bằng danh sách liên kết.



3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

- Các phần tử của hàng đợi được lưu trữ lần lượt bằng các phần tử của mảng;
- Biến kiểu nguyên: T để chỉ vị trí phần tử tiếp theo sẽ được thêm vào hàng đợi;
- Biến kiểu nguyên: H lưu trữ vị trí của phần tử đứng đầu hàng đợi;
- Biến kiểu nguyên: S lưu số phần tử hiện thời có trong hàng đợi;
- Biến: `kich_thuoc` lưu kích thước của mảng.

Giả sử các phần tử của ngăn xếp có kiểu là ELEMENT (ELEMENT có thể là NGUYÊN, KÝ TỰ, THỰC, PHỨC,...), kiểu dữ liệu hàng đợi với dung lượng lưu trữ tối đa là 10 phần tử và được cài đặt bằng mảng có thể được định nghĩa như sau:

```
typedef int ELEMENT;
struct QUEUE_ARRAY
{
    ELEMENT ele[10];
    int capacity, H, T, S;
};
```

CÁU TRÚC DỮ LIỆU VÀ GIẢI THUẬT



3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

Khởi tạo hàng đợi sẽ được viết như sau:

```
QUEUE_ARRAY INIT_QUEUE_ARRAY(QUEUE_ARRAY q)
{
    q.capacity=10;
    q.H=-1;
    q.T=-1;
    q.S=0;
    return q;
}
```



3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

- Giải thuật để kiểm tra trạng thái của hàng đợi rỗng:

```
int IS_EMPTY(QUEUE_ARRAY q)
{
    int empty;
    empty=0; if
    (q.S==0)
        empty=1;
    else
        empty=0;
    return
        empty;
}
```

- Giải thuật để kiểm tra trạng thái của hàng đợi đầy:

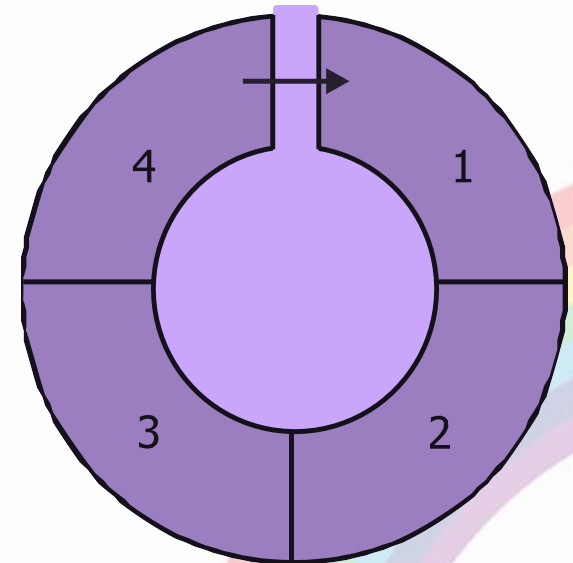
```
int IS_FULL(QUEUE_ARRAY q)
{
    int full;
    full=0;
    if (q.S==q.capacity)
        full=1;
    else
        full=0;
    return
        full;
}
```



3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

Ta hãy hình dung mảng giống như 1 vòng tròn, các phần tử sẽ được xếp thứ tự tự theo chiều kim đồng hồ:

- Phần tử cuối cùng của mảng sẽ được nối với phần tử đầu tiên của mảng;
- Khi đó, trong trường hợp $T = \text{kich_thuoc}$, nếu ta thêm một phần tử vào hàng đợi, phần tử này sẽ được xếp vào vị trí cuối cùng của mảng;
- Có trường hợp phần tử đứng đầu hàng đợi được xếp ở cuối mảng hay $H = \text{kich_thuoc}$, còn các phần tử đứng sau lại được xếp trên đầu mảng.





3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

Và giải thuật rút phần tử ra khỏi hàng đợi như sau:

```
ELEMENT DEQUEUE(Queue_ARRAY q)
{ ELEMENT e;
  if (IS_EMPTY(q)!=0)
    printf("hang doi rong khong the lay phan tu ra");
  else
  {   e = q.ele[q.H];
      if (q.H==q.T){           q.H=-1;           q.T=0;}
      else                   q.H=q.H+1;
      q.S=q.S-1;
  }
  return e;
}
```




3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

Giải thuật thêm phần tử vào hàng đợi như sau:

```
void ENQUEUE(Queue_ARRAY q, ELEMENT e)
{
    if (IS_FULL(q) != 0)
        printf("hàng đợi đầy không thể chen thêm");
    else
    {
        • if (q.T == q.capacity - 1) q.T = 0;
        • else
            • q.T = q.T + 1; q.ele[q.T] = e; q.S = q.S + 1;
    }
}
```



3.1. CÀI ĐẶT HÀNG ĐỢI BẰNG MẢNG

- Ưu điểm của phương pháp cài đặt hàng đợi bằng mảng:
 - Thời gian truy xuất nhanh;
 - Các thao tác trên hàng đợi có thời gian thực hiện nhanh.
- Nhược điểm: kích thước của hàng đợi bị giới hạn (không được vượt quá kích thước của mảng)



3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

- Mỗi phần tử của danh sách liên kết phải có:
 - 1 trường để lưu thông tin về phần tử trong hàng đợi (kiểu ELEMENT);
 - 1 con trỏ để chỉ tới phần tử tiếp theo trong danh sách.
- Do đó ta có thể định nghĩa kiểu cho các phần tử trong danh sách như sau:

```
typedef int ELEMENT;  
struct LIST_ELEMENT  
{  
    ELEMENT info;  
    struct LIST_ELEMENT *next;  
};
```



3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

- Sử dụng một con trỏ H trỏ tới phần tử đầu tiên của danh sách.
- Nếu danh sách rỗng, không có phần tử nào trong hàng đợi, con trỏ đó sẽ có giá trị RỖNG (NULL).
- Biến S kiểu NGUYÊN sử dụng để lưu số phần tử hiện thời trong hàng đợi.

```
struct QUEUE_LIST
{
    struct LIST_ELEMENT *H;
    int S; info;
};
```



3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

Khởi tạo một hàng đợi sử dụng danh sách liên kết:

```
QUEUE_LIST INITIALIZE(QUEUE_LIST q)
{
    q.H = NULL;
    q.S = 0;
    return q;
}
```



3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

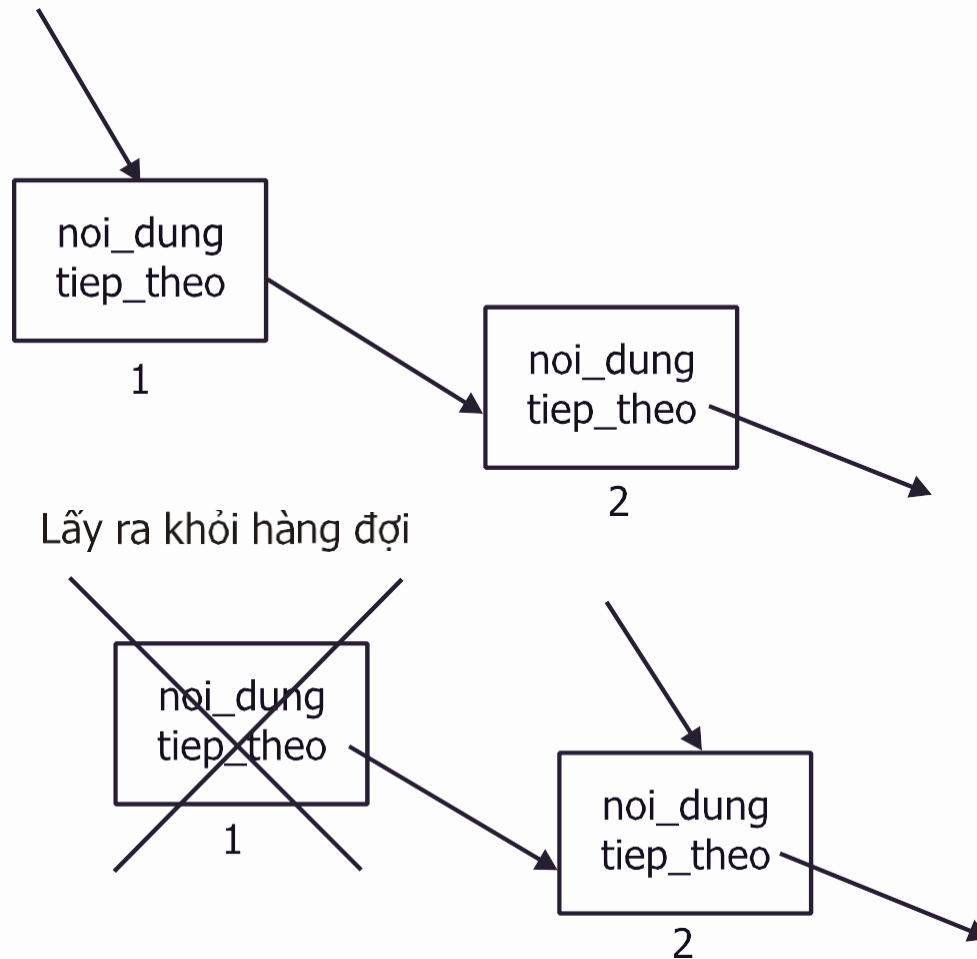
Giải thuật thêm phần tử vào hàng đợi :

```
QUEUE_LIST ENQUEUE(QUEUE_LIST q,ELEMENT e)
{
    LIST_ELEMENT *p,*m; m
    =new LIST_ELEMENT;
    m->info =e;
    m->next =NULL;
    if (q.S== 0)                q.H=m;
    else
    {
        p = q.H;
        for (int i=0;i<q.S-2;i++) p = p-
            >next;
        p->next=m;    }
    return q;
}
```



3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

Thao tác lấy ra khỏi hàng đợi (dequeue).





3.2. CÀI ĐẶT HÀNG ĐỢI BẰNG DANH SÁCH LIÊN KẾT

Thuật toán trên có sử dụng một câu lệnh tương đối phức tạp là $q.H \rightarrow info$. Câu lệnh trên được giải thích như sau:

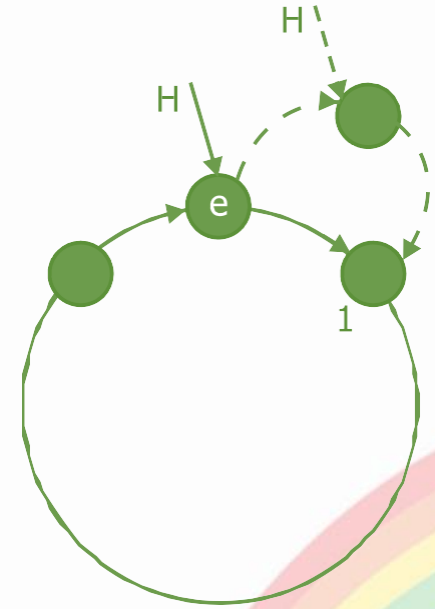
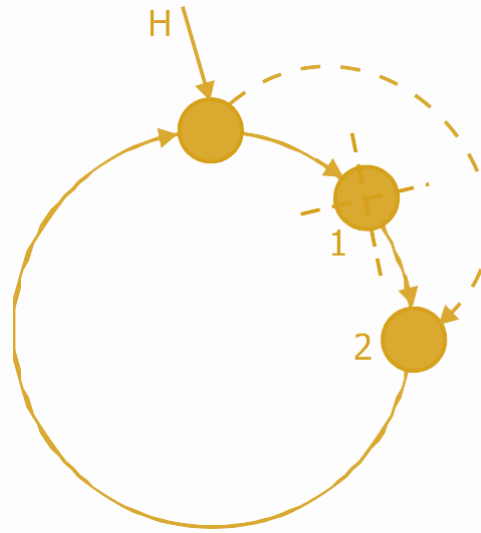
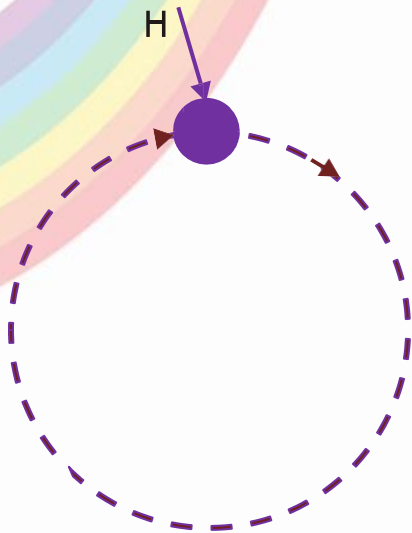
- $q.H$ để tham chiếu tới con trỏ H của hàng đợi q ;
- $q.H$ thuộc kiểu CON TRỎ tới `LIST_ELEMENT`, vì vậy để truy xuất tới các trường của bản ghi được trỏ bởi $q.H$ chúng ta sử dụng phép toán tham chiếu \rightarrow ;
- $q.H \rightarrow info$: để lấy về giá trị của trường `info` trong bản ghi được trỏ bởi $q.H$;
- $q.H \rightarrow next$: để lấy về giá trị của trường `next` trong bản ghi được trỏ bởi $q.H$.



4. HÀNG ĐỢI VÒNG TRÒN



4. HÀNG ĐỢI VÒNG TRÒN



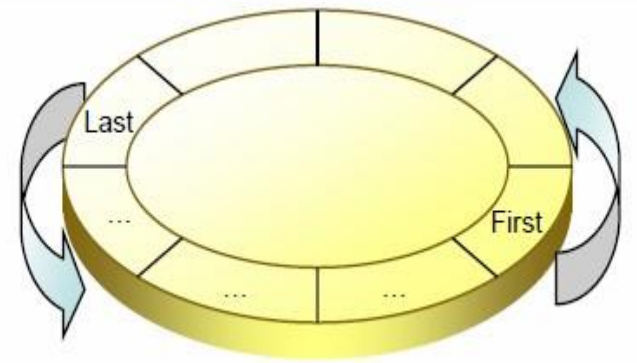
Danh sách vòng (biểu diễn bằng mảng hoặc danh sách liên kết): các phần tử của hàng đợi được xếp quanh vòng theo một hướng nào đó.



4. HÀNG ĐỢI VÒNG TRÒN

Nghĩa là:

- Các phần tử nằm trên phần cung tròn từ vị trí H tới vị trí T là các phần tử của hàng đợi;
- Một biến S lưu số phần tử trong hàng đợi;
- Hàng đợi rỗng hoặc đầy khi H bằng T;
- Việc thêm một phần tử vào hàng đợi tương đương với việc ta dịch chỉ số T theo vòng một vị trí rồi đặt giá trị mới vào đó;
- Việc loại bỏ một phần tử trong hàng đợi tương đương với việc lấy ra một phần tử tại vị trí H rồi dịch chỉ số H theo vòng.



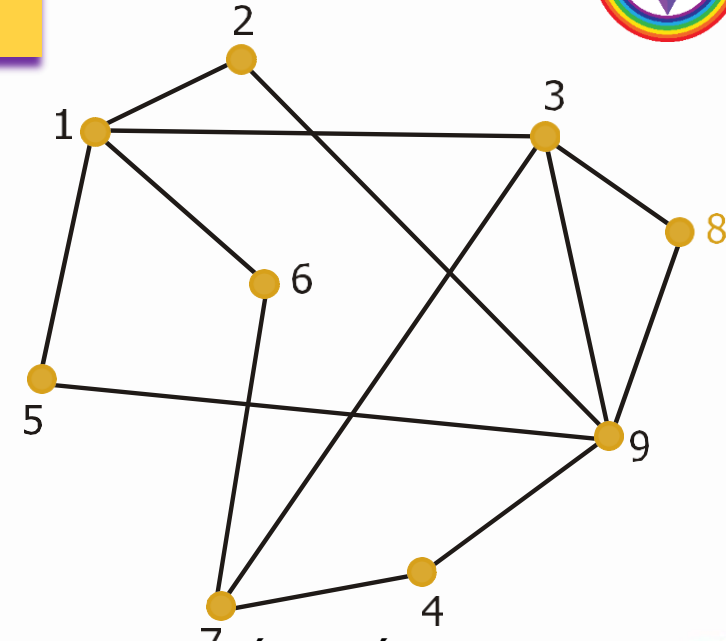


5. ỨNG DỤNG CỦA HÀNG ĐỢI



5. ỨNG DỤNG CỦA HÀNG ĐỢI

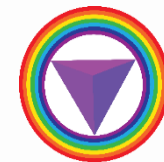
- Được dùng nhiều trong thực tế;
- Một trong những ứng dụng thường gặp nhất là trong toán học đồ thị: duyệt đồ thị theo chiều sâu và chiều rộng.
- **Ví dụ:** Có một đồ thị được cho trong hình... Vấn đề đặt ra là tìm đường đi ngắn nhất từ đỉnh 2 đến đỉnh 7.



Bằng việc sử dụng hàng đợi, ta có thể tìm được đường đi ngắn nhất từ 2 đến 7 bằng thuật toán duyệt đồ thị theo chiều rộng:

- Bước 1: Khởi tạo một hàng đợi Q rỗng;
- Bước 2: Xếp phần tử 2 vào hàng đợi Q;
- Bước 3: Lần lượt lấy các phần tử ra khỏi hàng đợi Q và xếp tất cả các đỉnh kề với nó (trừ những đỉnh đã xét) vào hàng đợi cho đến khi gặp được phần tử có giá trị bằng 7.

Nếu không còn phần tử nào trong hàng đợi mà chưa tìm được phần tử có giá trị bằng 7 thì kết luận không có đường đi từ 2 đến 7.



5. ỨNG DỤNG CỦA HÀNG ĐỢI

Kết quả của thuật toán trên được mô tả trong bảng sau:

Thao tác	Kết quả Q
Xếp vào hàng đợi	2
Lấy 1 phần tử ra khỏi hàng đợi: 2	Rỗng
Xếp tất cả các phần tử kề với 2 vào Q	1, 9
Lấy 1 phần tử ra khỏi hàng đợi: 1	9
Xếp tất cả các phần tử kề với 1 vào Q	9 3 5 6
Lấy 1 phần tử ra khỏi hàng đợi: 9	3 5 6
Xếp tất cả các phần tử kề với 9 vào Q	3 5 6 4 8 (2 và 5 đã được xét nên không được xếp vào)
Lấy 1 phần tử ra khỏi hàng đợi: 3	5 6 4 8
Xếp tất cả các phần tử kề với 3 vào Q	5 6 4 8 7



5. ỨNG DỤNG CỦA HÀNG ĐỢI

Như vậy sau 9 thao tác ta đã gặp được phần tử có giá trị bằng 7. Lần ngược lại các thao tác ta sẽ thu được 1 đường đi ngắn nhất từ 2 đến 7:

- 7 được xếp vào Q do liền kề với 3;
- 3 được xếp vào Q do liền kề với 1;
- 1 được xếp vào Q do liền kề với 2.

Vậy đường đi ngắn nhất từ 2 đến 7 tìm được là: 2 1 3 7.



TÓM LƯỢC CUỐI BÀI

Hàng đợi còn gọi là danh sách hạn chế. Nên các bạn cần lưu ý một số vấn đề sau:

- Trình bày đúng khái niệm về hàng đợi. Phân biệt điểm giống và khác giữa hàng đợi và danh sách;
- Mô tả đúng các đặc tả về hàng đợi và các phương án cài đặt cho hàng đợi;
- Trình bày được các ứng dụng của hàng đợi và vận dụng hàng đợi vào giải quyết các bài toán.