

VÕ TIỀN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Cấu Trúc Dữ Liệu và Giải Thuật (DSA)

DSA3 - HK242

Cuối Kỳ

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>

Mục lục

1	Lý Thuyết Hash và Search	2
2	Câu hỏi trong các đề thi	3



1 Lý Thuyết Hash và Search

Các phương pháp xử lý đụng độ trong bảng băm

1. Phương pháp kết chuỗi (Chaining)

- Mỗi ô trong bảng băm chứa một danh sách liên kết (hoặc một cấu trúc dữ liệu tương tự).
- Khi có nhiều phần tử cùng chỉ số băm, chúng sẽ được thêm vào danh sách đó.
- **Ưu điểm:** Không giới hạn số lượng phần tử, dễ cài đặt.
- **Nhược điểm:** Tốn bộ nhớ vì phải dùng danh sách liên kết.

2. Phương pháp địa chỉ mở (Open Addressing)

- Khi xảy ra đụng độ, ta tìm một ô trống khác trong bảng băm.
- Không dùng danh sách liên kết, tất cả phần tử được lưu trong bảng băm.

Dò tìm tuyến tính (Linear Probing)

- Nếu ô đã bị chiếm, thử ô kế tiếp ($\text{index} + 1$).
- Nếu vẫn bị chiếm, tiếp tục dò ($\text{index} + 2, \text{index} + 3, \dots$).
- **Nhược điểm:** Dễ bị **gom cụm chính cấp** (nhiều phần tử liên tiếp nhau, làm giảm hiệu suất).

Dò tìm bậc hai (Quadratic Probing)

- Nếu ô đã bị chiếm, thử ô ở khoảng cách tăng dần theo bậc hai:

$$\text{index} + 1^2, \text{index} + 2^2, \text{index} + 3^2, \dots$$

- Giảm bớt gom cụm chính cấp nhưng vẫn có thể bị **gom cụm thứ cấp**.

Băm đôi (Double Hashing)

- Khi xảy ra đụng độ, dùng một **hàm băm thứ hai** để tìm vị trí mới.
- Công thức:

$$\text{new_index} = (\text{index} + i \times \text{hash2}(\text{key})) \bmod \text{table_size}$$

- **Ưu điểm:** Hạn chế tối đa hiện tượng gom cụm.

Hiện tượng gom cụm (Clustering)

1. Gom cụm chính cấp (Primary Clustering)

- Xảy ra khi nhiều phần tử bị đụng độ và dồn về một nhóm lớn (do Linear Probing gây ra).
- Làm giảm hiệu suất vì các phần tử mới cũng sẽ bị chen vào cùng nhóm đó.

2. Gom cụm thứ cấp (Secondary Clustering)

- Xảy ra trong Quadratic Probing và các phương pháp băm có quy luật cố định.
- Nhóm phần tử bị xếp gần nhau vì cùng sử dụng chung công thức tìm vị trí.

Hãy vẽ bảng băm ra giấy để hình dung dễ hơn, nhất là khi làm bài toán Insert/Delete.

Tìm kiếm (Search)

- **Interpolation Search:** Tìm kiếm trên mảng đã sắp xếp, nhanh hơn Binary Search trong một số trường hợp.
- **Binary Search:** Có 2 cách cài đặt:
 - Đệ quy (Recursion)
 - Vòng lặp (for)



2 Câu hỏi trong các đề thi

- Dùng giải thuật tìm kiếm nội suy (interpolation search) để tìm vị trí của 48 trong dãy sau: [1,5,8,11,19,22,31,35,40,45,48,49,50]. Cần phải thực hiện bao nhiêu lần so sánh?
 - 2
 - 3
 - 4
 - 5
- Với mỗi dãy khoá dưới đây, lần lượt thêm vào một bảng băm (rỗng) có kích thước là 19, với hàm băm là $h(k) = k \% 19$, dãy khoá nào xảy ra ít đụng độ nhất?
 - 0, 19, 57
 - 19, 20, 21, 39, 40
 - 19, 20, 21, 22, 23, 40, 41, 57
 - 19, 20, 21, 22, 23, 46, 47, 57
- Trong các hiện thực hàm thăm dò hash $hp(k, p, M)$ với k là khoá và M là kích thước bộ nhớ, hiện thực nào tốt nhất để giải quyết đụng độ theo phương pháp đánh địa chỉ mở?
 - $h(k, p, M) = (a * k + b * p) \% M$, với a, b là hai hằng nguyên tố
 - $h(k, p, M) = h_1(k, M) + p * h_2(k, M)$, với h_1, h_2 là hai hàm hash
 - $h(k, p, M) = (h_1(k, M) + p * h_2(k, M)) \% M$, với h_1, h_2 là hai hàm hash
 - $h(k, p, M) = \text{rot}(k, p) \% M$, với rot là hàm xoay khoá p lần
 - Tất cả các câu trên đều sai
- Cho hàm hash như sau: $h(k, M) = \text{rot}(k, 2) \% M$, trong đó $\text{rot}(k, m)$ là phép quay sang trái m chữ số, M là kích thước bộ nhớ. Hãy cho biết kết quả của đoạn lệnh sau: `vector<int> v{1026, 1023, 1022, 1025, 1001}; for (auto val:v) cout << h1(val, 107) << ' ';`
 - 26 23 22 25 11
 - 42 63 70 49 3
 - 61 31 21 51 10
 - 610 310 210 510 011
- Khoá cần hash là một chuỗi ký tự. Hàm hash được định nghĩa như sau: tổng của ký tự hiện tại nhân với phần dư (khi chia 7) của ký tự ngay phía sau nó, tất cả lấy phần dư chia cho M , với M là kích thước bộ nhớ. Cho hàm `int h2(string str, int M)`, hiện thực bên trong là:
 - `int acc = 0; for (char c:str) acc += c; return acc % M;`
 - `for (char c:str) c += c; return c % M;`
 - Hai đáp án trên đều sai
 - Hai đáp án trên đều đúng
- Lần lượt thêm các khoá 5; 7; 12; 25; 36; 58 vào một bảng băm có kích thước là 11, sử dụng hàm băm $h(k) = k \bmod 11$. Nếu đụng độ xảy ra thì phương pháp giải quyết đụng độ được chọn là mở rộng địa chỉ (open addressing) với hàm dò tìm là $p(k, i) = h(k) + 2 * i + 1$. Vị trí của khoá 58 trong bảng băm là:
 - 0
 - 3
 - 6
 - 9
- Phát biểu nào sau đây là đúng về Quadratic probing?
 - Là một phương pháp hiện thực hàm băm.
 - Có thể dẫn đến Primary clustering.
 - Có thể dẫn đến Secondary clustering.
 - Có thể dẫn đến Linear clustering và Secondary clustering.
- Với mỗi dãy khoá dưới đây, lần lượt thêm vào một bảng băm (rỗng) có kích thước là 19, với hàm băm là $h(k) = k \% 19$, dãy khoá nào xảy ra ít đụng độ nhất?
 - 0, 19, 57
 - 19, 20, 21, 22, 23, 46, 47, 57
 - 19, 20, 21, 22, 23, 40, 41, 57
 - 19, 20, 21, 39, 40
- Phát biểu nào sau đây là đúng về tình trạng đụng độ trong phương pháp băm?
 - Đụng độ là tình trạng 2 khoá khác nhau cho ra 2 giá trị băm khác nhau.
 - Đụng độ là tình trạng 2 khoá giống nhau cho ra 2 giá trị băm khác nhau.
 - Đụng độ là tình trạng 2 khoá giống nhau cho ra 2 giá trị băm giống nhau.



- d) Đụng độ là tình trạng 2 khoá khác nhau cho ra 2 giá trị băm giống nhau. clustering.
10. Hãy xem xét một bảng băm có kích thước $m = 100$ và hàm băm $h(k) = \text{floor}(m(kA\%1))$ cho $A = \text{frac}(\text{sqrt}5 - 1)/2 = 0.618033$ Tính toán vị trí mà khóa $k = 123456$ được đặt trong bảng băm.
- a) 77 b) 89 c) 88 d) 82
11. Chèn các ký tự của chuỗi K R P C S N Y T J M vào bảng băm có kích thước 10. Sử dụng hàm băm $h(x) = (\text{ord}(x) - \text{ord}('a')) + 1\%10$ Nếu thăm dò tuyến tính được sử dụng để giải quyết va chạm, thì phần chèn sau gây ra va chạm
- a) Y b) P c) M d) C
12. Hãy xem xét một bảng băm có 100 vị trí. Các Collisions được giải quyết bằng cách sử dụng chaining. Giả sử băm đồng nhất đơn giản, xác suất để 3 vị trí đầu tiên không được lấp đầy sau 3 lần chèn đầu tiên là bao nhiêu?
- a) $(97 * 96 * 95)/100^3$ b) $(97 * 96 * 95)/(3! * 100^3)$
c) $(99 * 98 * 97)/100^3$ d) $(97 * 97 * 97)/100^3$
13. Một bảng băm có kích thước 11 phần tử sử dụng phương pháp địa chỉ mở để giải quyết đụng độ, trong đó phương pháp dò tìm bậc hai được gọi sử dụng. Biết rằng, hàm băm được sử dụng trong lần đầu tiên là $h'(k) = k\%11$ và phương pháp dò tìm bậc hai là $h(k, i) = (h'(k) + 2i^2)\%11$. Lần lượt thêm 38, 35, 28, 45, 94, 71, 40, 80 vào một bảng băm nói trên. Tổng các số trên 5 khe cuối cùng có chứa khóa là:
- a) 100 b) 110 c) 120 d) 130
14. Cho một bảng băm có kích thước $m = 19$ sử dụng một hàm băm $h(k) = k \bmod 19$. Lần lượt thêm các khóa 15, 78, 56, 25, 19, 38, 57, 76, 34, 53, 72, 91 vào bảng băm trên với các giả định: **Sử dụng phương pháp kết chuỗi (chaining) để giải quyết đụng độ.**, đếm tổng số tại vị trí bị đụng độ nhiều nhất:
- a) 3 b) 4 c) 5 d) 6
15. Phương pháp nào dưới đây có khả năng giảm thiểu hiệu quả hiện tượng gom cụm chính cấp (Primary Clustering) khi sử dụng địa chỉ mở để giải quyết đụng độ?
- a) Phương pháp dò tìm tuyến tính b) Phương pháp dò tìm bậc hai
c) Phương pháp băm đôi (Double Hashing) d) Phương pháp kết chuỗi (Chaining)
16. Khi sử dụng phương pháp dò tìm tuyến tính trong bảng băm với hàm băm $h(k) = k \% 19$ và bảng băm có kích thước $m = 19$, hiện tượng gom cụm thứ cấp (Secondary Clustering) sẽ xảy ra trong trường hợp nào?
- a) Khi tất cả các phần tử đều có giá trị băm giống nhau
b) Khi các phần tử có giá trị băm gần nhau và liên tiếp bị đụng độ
c) Khi các phần tử có giá trị băm phân tán đều trong bảng
d) Không có hiện tượng gom cụm thứ cấp xảy ra
17. Với phương pháp băm đôi (Double Hashing), khi hàm băm thứ hai $h_2(k) = 1 + (k\%(m-1))$, phương pháp này giúp gì trong việc giải quyết đụng độ so với các phương pháp dò tìm tuyến tính hoặc bậc hai?
- a) Giảm đáng kể hiện tượng gom cụm và phân tán các phần tử
b) Chỉ giảm được gom cụm thứ cấp, không ảnh hưởng gom cụm chính cấp
c) Tăng cường khả năng tìm kiếm các phần tử đã bị đụng độ
d) Không có ảnh hưởng đến hiện tượng gom cụm
18. Khi sử dụng phương pháp địa chỉ mở (Open Addressing) với phương pháp dò tìm bậc hai, với công thức $h(k, i) = (h(k) + i^2)\%m$, trong trường hợp bảng băm có kích thước $m = 7$, khóa 13 sẽ được lưu tại vị trí nào sau khi thêm lần lượt các khóa: 13, 27, 14, 10, 6?
- a) Vị trí 2 b) Vị trí 3 c) Vị trí 4 d) Vị trí 5



19. Phương pháp nào sau đây có khả năng giảm thiểu sự phụ thuộc vào kích thước bảng băm khi sử dụng trong một hệ thống có thể thay đổi kích thước bảng theo thời gian?

- a) Phương pháp băm đôi (Double Hashing) b) Phương pháp kết chuỗi (Chaining)
c) Phương pháp dò tìm bậc hai (Quadratic Probing) d) Phương pháp dò tìm tuyến tính (Linear Probing)

[6.3] Giải thuật sau tìm kiếm target trong mảng tăng dần a: Với $n = 13$ và $\text{target} > a[n-1]$, hãy cho biết số lần thực hiện phép so sánh trong các câu lệnh if

```
1 int i , low = 0 , high = n - 1 ;
2 while ( low < high ) {
3     i = ( low + high ) / 2 ;
4     if ( a[i] == target ) return i ;
5     else if ( a[i] < target ) low = i + 1 ;
6     else high = i - 1 ;
7 }
8 return -1;
```

- a) 4 b) 6 c) 7 d) 8

20. Cho foo3 là giải thuật tìm kiếm nội suy (Interpolation Search):

```
1 int foo3(int *arr, int n, int x) {
2     int i1 = 0, i2 = (n - 1), pos;
3     while (i1 <= i2 && x >= arr[i1] && x <= arr[i2]) {
4         if (i1 == i2) {
5             if (arr[i1] == x) return i1;
6             return -1;
7         }
8         //Line 1
9         if (arr[pos] == x) return pos;
10        if (arr[pos] < x) i1 = pos + 1;
11        else i2 = pos - 1;
12    }
13    return -1;
14 }
```

Câu lệnh phù hợp để điền vào vị trí Line 1:

- (a) $\text{pos} = i1 + \text{float}(i2 - i1) * (x - \text{arr}[i1]) / (\text{arr}[i2] - \text{arr}[i1]);$
(b) $\text{pos} = i1 + \text{float}(i2 - i1) * (x - \text{arr}[i1]) / (\text{arr}[i1] - \text{arr}[i2]);$
(c) $\text{pos} = i1 + \text{float}(i2 - i1) * (\text{arr}[i1] - x) / (\text{arr}[i1] - \text{arr}[i2]);$
(d) $\text{pos} = i1 + \text{float}(i2 - i1) * ((x - \text{arr}[i1]) / (\text{arr}[i2] - \text{arr}[i1]));$
a) Có đúng 1 câu lệnh có thể chọn b) Có đúng 2 câu lệnh có thể chọn
c) Có đúng 3 câu lệnh có thể chọn d) Không có câu lệnh nào phù hợp

21. Cho hàm sau:



```
1 int fun1(int arr[], int size, int key, int k) {
2     int s = 0, e = size - 1, mid = s + (e - s) / 2;
3     int ans = -1;
4     while (s <= e) {
5         if (arr[mid] == key) {
6             if (arr[mid - k] == arr[mid]) e = mid - 1;
7             else if (arr[mid - (k - 1)] == arr[mid]) {
8                 ans = mid;
9                 break;
10            } else s = mid + 1;
11        } else if (key < arr[mid]) e = mid - 1;
12        else if (key > arr[mid]) s = mid + 1;
13        mid = s + (e - s) / 2;
14    }
15    return ans;
16 }
```

Hãy cho biết kết quả xuất ra màn hình sau khi đoạn mã dưới đây được thực thi.

```
1 int arr[]{1, 5, 5, 11, 19, 22, 22, 22, 22, 45, 48, 48, 48};
2 cout << fun1(arr, 13, 22, 4);
```

a) 5

b) 6

c) 7

d) 8

22. Cho hàm sau:

```
1 int foo1(int *arr, int idx1, int idx2, int x) {
2     if (idx2 >= idx1) {
3         int idx3 = idx1 + (idx2 - idx1) / 2;
4         if (arr[idx3] == x) return idx3;
5         if (arr[idx3] < x) return foo1(arr, idx1, idx3 - 1, x);
6         return foo1(arr, idx3 + 1, idx2, x);
7     }
8     return -1;
9 }
```

Hãy cho biết kết quả xuất ra màn hình sau khi đoạn mã sau thực thi xong.

```
1 int arr[]{2, 3, 4, 10, 40};
2 cout << foo1(arr, 0, 5, 10);
```

a) 0

b) 3

c) 4

d) Tất cả các đáp án trên đều sai

Câu hỏi thời còn thi tự luận và điền chỗ trống có thể luyện tay



23. Cho một bảng băm có kích thước $m = 19$ sử dụng một hàm băm $h(k) = k \bmod 19$. Lần lượt thêm các khóa 15, 78, 56, 25, 19, 38, 57, 76, 34, 53, 72, 91 vào bảng băm trên với các giả định: **Sử dụng phương pháp địa chỉ mở (open addressing) để giải quyết đụng độ với phương pháp dò tìm tuyến tính**: $hp(k, i) = (h(k) + i) \% m$, liệt kê bảng băm:
24. Cho một bảng băm có kích thước $m = 19$ sử dụng một hàm băm $h(k) = k \bmod 19$. Lần lượt thêm các khóa 15, 78, 56, 25, 19, 38, 57, 76, 34, 53, 72, 91 vào bảng băm trên với các giả định: **Sử dụng phương pháp địa chỉ mở (open addressing) để giải quyết đụng độ với phương pháp dò tìm bậc hai**: $hp(k, i) = (h(k) + i^2) \% m$, liệt kê bảng băm:
25. Cho một bảng băm có kích thước $m = 19$ sử dụng một hàm băm $h(k) = k \bmod 19$. Lần lượt thêm các khóa 15, 78, 56, 25, 19, 38, 57, 76, 34, 53, 72, 91 vào bảng băm trên với các giả định: **Sử dụng phương pháp địa chỉ mở (open addressing) để giải quyết đụng độ với phương pháp băm đôi (double hashing)**: $hp(k, i) = (h_1(k) + ih_2(k)) \% m$, $h_1(k) = h(k)$, $h_2(k) = 1 + (k \% 17)$, liệt kê bảng băm:
26. Cho dãy các số nguyên được sắp xếp tăng dần: 0, 1, 1, 2, 3, 5, 8, 13, 21, 55, 77, 89, 101, 201, 256, 780. Sử dụng Jump search để tìm khóa 8, số lần phải thực hiện phép so sánh biết rằng kích thước một block là 4
27. Cho dãy: [1, 5, 7, 8, 16, 30, 33, 35, 41, 49, 51, 52, 59, 61, 65, 68, 69, 71, 72, 75, 76, 83, 85, 89, 97]. Sử dụng tìm kiếm nội suy để tìm số 49 trong dãy trên. Số vị trí dò tìm đã đi qua cho đến khi dừng là:
28. Cho một dãy các số nguyên đã sắp xếp: [2, 9, 11, 14, 16, 18, 23, 25, 44, 49, 50, 53, 57, 70, 74, 76, 78, 79, 81, 83, 84, 86, 91, 93, 97]. Sử dụng giải thuật tìm kiếm nhị phân để tìm phần tử 84, số phép so sánh phải thực hiện là bao nhiêu?
29. Cho một dãy các số nguyên đã sắp xếp: [2, 9, 11, 14, 16, 18, 23, 25, 44, 49, 50, 53, 57, 70, 74, 76, 78, 79, 81, 83, 84, 86, 91, 93, 97]. Sử dụng giải thuật tìm kiếm Jump để tìm phần tử 84 với kích thước một block là 5, số phép so sánh phải thực hiện là bao nhiêu?
30. Cho một dãy các số nguyên đã sắp xếp: [2, 9, 11, 14, 16, 18, 23, 25, 44, 49, 50, 53, 57, 70, 74, 76, 78, 79, 81, 83, 84, 86, 91, 93, 97]. Hãy so sánh kết quả của câu 2 với việc sử dụng kích thước một block là 3 và 7. Liệu có tồn tại một kích thước nào mà số phép so sánh ít hơn khi dùng kích thước block là 5?
31. Cho một dãy các số nguyên đã sắp xếp: [2, 9, 11, 14, 16, 18, 23, 25, 44, 49, 50, 53, 57, 70, 74, 76, 78, 79, 81, 83, 84, 86, 91, 93, 97]. Sử dụng giải thuật tìm kiếm nội suy để tìm kiếm phần tử 84 trong danh sách trên. So sánh kết quả với việc tìm kiếm bằng giải thuật tìm kiếm nhị phân?