

Hash Search

🕒 작성일시	@2022년 8월 3일 오후 6:14
📌 강의 번호	
📅 Date	@2022년 7월 26일
🔗 자료	
☑ 복습	<input type="checkbox"/>

- Hash Search

선형 탐색이나 이진탐색의 전제 조건은 어떤 데이터가 어떤 요소에 들어 있는지 전혀 모르는 상태에서 검색을 시작했다. 그러나 해시탐색은 데이터의 내용과 저장한 곳의 요소를 미리 연계해 둬으로써 극히 짧은 시간안에 탐색 할수 있도록 고안된 알고리즘이다.

해시 탐색은 '데이터를 데이터와 같은 첨자의 요소에 넣어두면 한 번에 찾을 수 있다는 아이디어'에서 시작한다.

확실하게 쉽고 빠르게 찾을 수 는 있지만 문제는 단 1개 즉 10000 데이터를 저장하려면 공간이 9999의 불필요한 공간을 준비해야 한다. 낭비가 심해진다.

따라서 좀 더 효율적으로 저장하기 위해서 일정한 계산을 하여 그 계산 결과값과 같은 첨자의 요소에 보관하는 방법을 고안하였다.

해시 탐색법으로 데이터를 보관하는 알고리즘

- 해시 함수은 데이터의 저장소의 첨자를 계산하는데 사용한다.
- 저장소의 첨자가 겹치는 것을 '충돌'이라고 한다.
- 충돌이 발생하면 옆의 빈자리에 보관한다.

해시 탐색법은 다른 알고리즘과는 달리

- 해시 함수로 데이터를 저장하는 알고리즘
- 해시 함수로 데이터를 검색하는 알고리즘

1. 데이터를 저장하는 알고리즘

- 배열을 2개 준비한다.
- 하나의 배열에는 저장할 데이터가 담겨있고 - 기본 데이터 배열 (초기값)
- 또 하나의 배열에는 실제 데이터를 보관한다. - 저장 데이터 배열

해시 함수를 사용하여 데이터를 저장할때 같은 나머지가 발생하게 되면 그 다음 인덱스의 위치에 저장하여 해결한다. 그런데 또다른 데이터가 이미 존재하면 다시 한번 다음 인덱스의 위치로 이동하여 저장한다.

실제 저장할 데이터의 배열의 개수는 데이터 갯수의 1.5배가 적당한 것으로 연구되어 있다. 너무 많은 배열을 준비하면 메모리의 소모가 발생하고 너무 적게 배열을 준비하면 충돌이 많이 발생하게 되어 불필요한 검색과정의 추가가 필요해진다.

0	1	2	3	4	5	6
---	---	---	---	---	---	---

0	15	23	3	11	26	6
---	----	----	---	----	----	---

현재 방의 갯수는 7개
어떤 수라도 7로 나누면 그 나머지는 언제나 0에서 6으로만 나온다.

$11 \% 7 = 4$
 $15 \% 7 = 1$
 $23 \% 7 = 2$
 $26 \% 7 = 5$

