Problem #5

1. q와 p를 시작점과 종료점으로 선택하여 재귀 호출로 문제를 해결한다. 그런 다음 호출된 함수 내에서 q와 p를 반으로 나누어 각각 다시 호출하는 과정을 반복하는데, q-p가 1보다 작을 때, 즉 찾고자 하는 배열의 원소가 하나 남았을때 그 값이 t 이면 1, 그렇지 않으면 0을 반환하여 함수를 종료한다.
2. 선형 탐색의 경우, 최선의 경우에는 1 (맨 앞에 있는 경우), 최악의 경우에는 n(맨 뒤에 있는 경우)의 시간이 소요되며, 분할 정복을 통한 경우 최선 또는 최악의 경우 모두 log n의 시간이 소요된다.

Problem #6

4^(n+1) -1 \* n^2 log n -> O(4^n)

Problem #7

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 1 | 2 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |

[ FIFO ]

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 5 | 5 | 5 | 5 | 4 | 4 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 5 |
|  |  | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 2 | 2 | 2 |
|  |  |  |  |  | 4 | 4 | 4 | 4 | 3 | 3 | 3 |

[ LRU ]

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 5 |
|  | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
|  |  | 3 | 3 | 3 | 3 | 5 | 5 | 5 | 5 | 4 | 4 |
|  |  |  |  |  | 4 | 4 | 4 | 4 | 3 | 3 | 3 |

FIFO 알고리즘의 페이지 폴트 횟수 = 10, LRU 알고리즘의 페이지 폴트 횟수 = 8

LRU 알고리즘이 더욱 효율적이다.