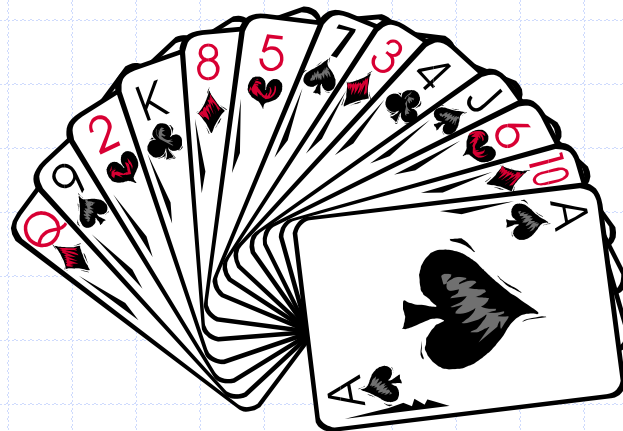
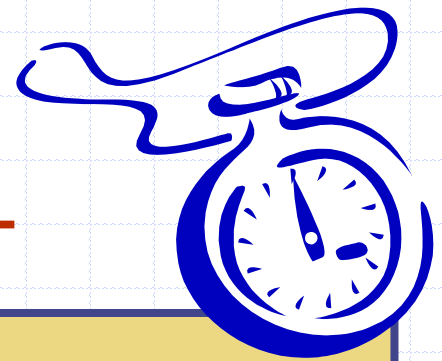


정렬 일반



Outline

- ◆ 9.2 정렬의 안정성
- ◆ 9.3 비교정렬 알고리즘 비교



비교정렬 알고리즘 비교

	시간	주요 전략	비고
선택 정렬	$O(n^2)$	◆ 우선순위 큐 (무순 리스트로 구현)	◆ 제자리 ◆ 느림 (소규모 입력에 적당)
삽입 정렬	$O(n^2)$	◆ 우선순위 큐 (순서 리스트로 구현)	◆ 제자리 ◆ 느림 (소규모 입력에 적당)
힙 정렬	$O(n \log n)$	◆ 우선순위 큐 (힙으로 구현)	◆ 제자리 ◆ 빠름 (대규모 입력에 적당)
합병 정렬	$O(n \log n)$	◆ 분할통치	◆ 순차 데이터접근 ◆ 빠름 (초대규모 입력에 적당)
퀵 정렬	$O(n \log n)$ 기대시간	◆ 분할통치	◆ 제자리, 무작위 ◆ 가장 빠름 (대규모 입력에 적당)



정렬의 안정성

- ◆ 키-원소 항목들을 정렬할 때, 중요한 이슈는 **동일 키가 어떻게 처리되는냐**는 것이다
- ◆ $L = ((k_0, e_0), \dots, (k_{n-1}, e_{n-1}))$ 을 항목들의 리스트라 하자
- ◆ 두 개의 항목 (k_i, e_i) 과 (k_j, e_j) 에 대해,
 - $k_i = k_j$ 며 정렬 전에 (k_i, e_i) 가 (k_j, e_j) 보다 앞서 있었다면(즉, $i < j$),
 - 만약 정렬 후에도 (k_i, e_i) 가 (k_j, e_j) 보다 앞서 있다면,
 - 그 정렬 알고리즘을 **안정적**(stable)이라고 말한다
- ◆ 정렬 알고리즘에 있어서 **안정성**(stability)은 중요 – 왜냐면 많은 응용에서 동일 키 원소들의 원래 순서가 보존되어야 할 필요가 있기 때문