정렬 알고리듬 이해를 위한 JAVA 애플릿 개발

최관순, 김진만, 최규오, 전홍구
순천향대학교 정보기술공학부

A Development of Java Applet for understanding Sort Algorism

Kwan-Sun Choi, Jin-Man Kim, Kyu-Oh Choi, Heung-Goo Jun
Division of Information Technology, Soonchunhyang Univ*
E-mail : cks1329@asan.sch.ac.kr, mani@gis-ip.sch.ac.kr

요 약

웹은 이제 우리 생활의 일부분이 되어 가고 있다. 많은 사람들이 웹을 이용하여 중요한 정보를 얻고 있으며, 이로 인하여 많은 콘텐츠들이 개발되고 있다. 그 중 학습을 목적으로 하는 교육 콘텐츠도 개발되고 있는데, 이러한 개발은 교육 및 훈련이 쉽고, 원하는 시간에 반복 학습이 가능하기 때문에 많이 이루어지고 있다. 본 연구에서는 알고리듬의 이해를 도와 주는 JAVA 프로그램을 개발하게 되었다. 웹에서의 구현을 위해 JAVA를 사용하였으며, 기존의 많은 알고리듬 중 정렬 알고리듬을 내용으로 하는 교육 콘텐츠를 개발하게 되였다.

1. 서론

정보통신의 발달과 함께 사회 환경의 변혁을 가져오고 있는 정보화 물결은 교육 환경에서도 예외는 아니다. 격변하는 정보화 사회의 요구에 부응하여 업무 수행에 필요한 신기술과 지식을 습득하는 동시에 직업 전환의 기회에 필요한 교육 및 훈련을 쉽게 제공받을 수 있는 수단으로서 웹을 이용한 가상 교육의 중요성은 날로 증가하고 있다. 정보화시대로 대표되는 현대 사회에서는 첨단 정보통신 기술의 발달로 기존의 강의실과 실습실로 대표되는 획일화된 닫힌 공간에서 벗어나 정보통신망을 활용한 열린 공간으로 학습장 개념이 확대되고 있다. 웹의 등장과 함께 급속한 컴퓨터 네트워크의 발달은 교사와 학습자가 면대면 방식으로 실시해온 전통적인 교육의 개념에서 벗어나 직접 대면하지 않고도 시간과 공간으로부터 자유로운 상태에서 다양한 통신 수단을 이용해 교수-학생간 학습이 이루어지고 있으며, 언제 어디서나 손쉽게 인터넷을 이용하여 교육을 받을 수 있게 되었다. 이러한 장점으로 웹을 이용한 가상 교육은 현재 계속 증가하고 있으며, 그 예로 현재 본 대학에서는 웹을 이용한 가상 대학을 운용하고 있고, 학생들로부터도 좋은 반응을 받고 있다.

본 연구에서 웹을 이용한 가상 학습 콘텐츠를 개발하였다. 프로그램 언어를 공부함에 있어 중요한 알고리듬에 대한 교육 학습 콘텐츠 개발이며, 웹으로 구현하기 위해 JAVA를 이용하였다. 그리고 많은 알고리듬들 중 학습 교재로 많이 이용되어지는 정렬 알고리듬을 내용으로 하는 교육 콘텐츠를 개발하게 되였다.

2. 본론

본 연구는 C언어에서 가장 많이 다루어지면서 비교적 복잡한 알고리즘 중에 하나인 정렬 알고리즘에 대한 교육 콘텐츠이다. 정렬 알고리듬 학습 애플릿은 웹에서 구현하기 위해 JAVA를 이용하였으며, 알고리듬의 이해를 돕기 위해 JAVA Thread에 의한 실시간 애니메이션 기법을 사용하였다. 그리고 정렬간의 비교

통계 자료를 통해 정렬 알고리듬 성능을 측정할 수 있게 하였다. 그림 1은 정렬 알고리듬 학습 애플릿의 메인 화면이다.

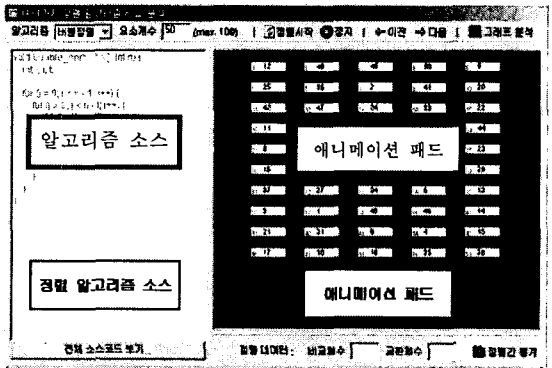

그림 1 정렬 알고리즘 학습 애플릿

본 정렬 애플릿에서는 버블정렬(bubble), 선택정렬(selection), 삽입정렬(insertion), 쉘정렬(shell), 퀵정렬(quick) 등 총 5가지의 정렬 알고리즘을 다룬다.

애플릿의 전체 구성은 맨 위쪽으로 툴바가 있으며 좌측으로 각 정렬 알고리듬을 C언어로 구현한 소스코드가 구성되어 지고, 우측으로는 애니메이션 패드로 구성 되여 있다. 이 패드는 정렬 할 값을 랜덤 하게 배열하였으며 학습자가 입력한 정렬 요소 개수에 따라서 1부터 정렬 요소 개수까지의 숫자가 랜덤 하게 배열하는 방법을 사용했다. 예를 들어 학습자가 10개의 정렬을 수행한다고 하였을 때, 애플릿은 1부터 10까지의 숫자를 무작위 순으로 배열하여 최초 화면에 보여준다.

```
private boolean generating = true;
        // 생성 플래그
private Integer randomInt = null;
        // 랜덤숫자를 임시로 저장
private Vector vector;
        // 검증된 랜덤숫자를 저장할 벡터
vector = new Vector();

for(int h = 0; h < padNum; h++) {
        // 일정범위의 숫자를 랜덤하게 생성
      while(generating){
              randomInt = new
Integer(((int)(Math.random()*padNum)+1));
            if(!vector.contains(randomInt))
                    randnum = null;
            else
                    generating = false;
      }
    vector.addElement(randnum);
    generating = true;
}
```

학습자는 애플릿 툴바에서 정렬 알고리즘을 선택하고 정렬에 사용될 숫자패드의 개수를 입력한 후 정렬 시작 버튼을 클릭 함으로써 정렬 과정을 수행하게 된다. (그림 2 참조)

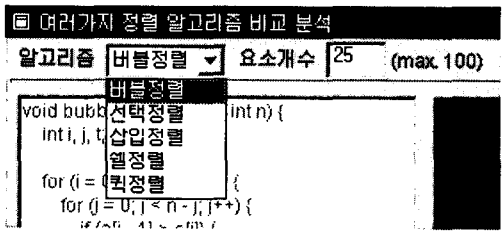

그림 2 알고리즘 선택과 개수 지정

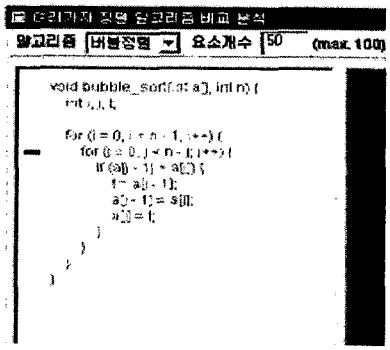

그림 3 소스 애니메이션

정렬간에는 소스 코드 부분에서 현재 선택되어진 정렬 소스가 보여지며 우측 숫자 패드 애니메이션에 맞추어 소스 부분에서도 진행되어 지고 있는 소스라인이 표시된다.(그림 3 참조) 우측 숫자 패드 애니메이션은 그림 4과 같은 모습을 보여준다.

정렬 애니메이션 과정은 그림에서 보는바와 같이 먼저 비교 대상 숫자 패드의 색이 짙은 노랑으로 반전되면서(a) 공간 확보를 위해 상하로 이동한다(b). (c)에서 패드의 교환이 이루어지며 마지막으로 상하로 이동한 뒤 패드의 색이 원래대로 돌아가면서 한번의 교환 과정을 마치게 되는 방식이다(d).

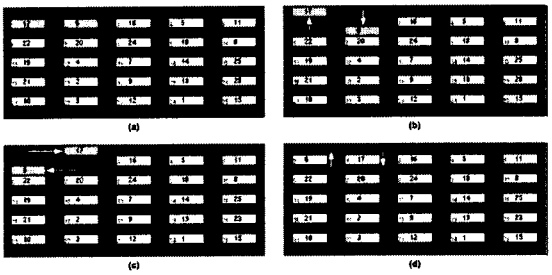

그림 4 숫자 패드 애니메이션 과정

위 애니메이션 과정이 숫자 패드로 숫자들이 어떻게 정렬되어 지는가를 보여 주었다면, 그림 5의 그래프 애니메이션을 통해 정렬 알고리듬의 패턴을 분석할 수 있게 하였다. 이 정렬 과정을 그래프로 보기 위해 툴바의 '그래픽 분석' 버튼을 누르면 새로운 프레임 창이 생성되면서 정렬 과정을 애니메이션 그래프로 보여준다. 그림 5는 하나의 예로써 버블정렬(bubble)에 대한 정렬 과정을 그래프로 본 화면이다.

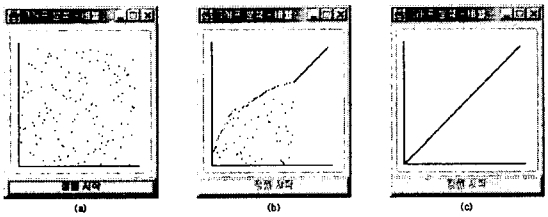

그림 5 정렬 그래프의 변화 과정

```
void bubble_sort(int a[], int n) {
    int i, j, t;
    for (i = 0; i < n - 1; i++) {
        for (j = 0; j < n - j; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```

그래프의 X축은 배열의 첨자 j를 의미하고 Y축은 배열 값 a[j]를 의미한다. 그래프 중 (a)는 정렬이 수행되기 전 초기상태이며 (b)는 1/2정도가 진행된 상태이고 (c)는 정렬이 완료되었을 때의 모습니다.

그래프를 보면 상단 부분이 먼저 정렬되는 모습이 보이고 또한 하단 부분은 조금씩 정렬이 되어가는 모습이 보인다. 이렇게 버블정렬은 최대 값을 찾는 것과 대충 정렬을 하는 두 가지 일을 하기 때문에 실행 시간이 상당히 느리다는 것을 알 수 있다.

정렬 작업이 끝나면 그림 6와 같이 애플릿의 하단에 2가지 정보가 나타난다. 정렬을 위해서 몇 번이나 값의 비교가 있었는지, 비교 후 실제로 교환이 이루어진 회수가 몇 번인지를 체크하고 이를 정렬별로 저장해 둔다.

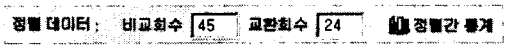

그림 6 정렬 간 비교 및 교환회수 정보 창

저장된 데이터는 나중에 각 정렬의 장단점을 비교하는 척도로 쓰이게 되는데 예를 들어 정렬 요소가 많고 적음에 따라 유리한 알고리즘을 가리거나, 정렬이 거의 이루어지지 않은 상태에서 정렬이 어느 정도 되어 있는 상태를 보고 유리한 알고리즘을 가려냄으로써 학생들로 하여금 언제 어떤 알고리즘을 적용하는 것이 가장 적절한지를 학습할 수 있게 하였다.

그림 7은 각 정렬간에 비교 회수와 교환 회수를 막대그래프와 수치로 보여주는 통계 화면이다.

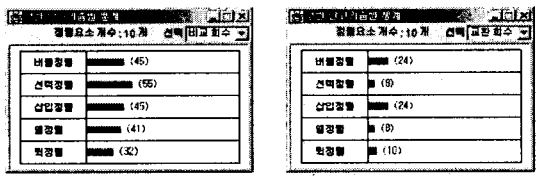

그림 7 정렬 알고리즘별 통계

위의 그림 6가 현재 진행중인 정보에 대한 비교 값과 교환 값을 보여준다. 하지만 이것으로는 모든 정렬을 비교할 수 없다. 그래서 그림 6에서 보여지는 '정렬간 통계' 버튼을 누르면 현재 선택되어진 같은 개수와 같은 값을 가지고 각 정렬들간의 비교 회수와 교환 회수를 비교할 수 있게 하였다.

표 1은 각 정렬에 대하여 요소가 10개, 50개, 100개일 때의 정렬 작업을 3번씩 수행해 그 평균 수치를 나타낸 것이다.

이 수치로 우리가 알 수 있는 것은 정렬 요소의 개수에 따른 알고리즘별 성능으로 각 조건마다 어떠한 알고리즘이 유리한가를 통계적으로 알 수 있게 해준다.

표 1 알고리즘 간 개수별 정렬 통계

요소개수	10		50		100	
	비교	교환	비교	교환	비교	교환
버블정렬	45	19.3	1225	611	4950	2508.7
선택정렬	55	8	1275	45	5050	96
삽입정렬	45	19.3	1225	611	4950	2508.7
쉘정렬	46.3	10.7	1171.7	186.3	4274	686
퀵정렬	33.3	9.3	343.7	69.7	830	159.3

표의 결과에서 볼 수 있듯이 본 통계에서는 정렬 요소의 개수가 적을 때는 쉘정렬이 가장 우수한 성능을 보여주었다. 반대로 정렬 요소의 개수가 많을 때는 단연 퀵정렬이 가장 우수한 성능을 보여주고 있음을 볼 수 있다. 하지만 이 결과는 최초 생성된 배열의 정렬 정도에 따라서 달라질 수 있는 문제이기 때문에 절대적인 통계는 될 수 없음을 밝힌다.

3. 결론

본 연구는 학생들이 웹을 이용하여 정렬 알고리듬을 학습시키는데 목적이 있다. 따라서 쉬운 학습 이해를 돕기 위해 애니메이션 기법을 사용하여 정렬 알고리듬들이 어떻게 동작되어 지는지를 학습할 수 있게 하였으며, 그래프 애니메이션을 통한 알고리듬들의 패턴 이해를 돕고 있다. 그리고 비교 회수와 교환 회수의 통계를 이용하여 학습자로 하여금 알고리듬 성능에 대한 이해를 돕고 있다.

이렇듯 다양한 모습을 이용하여 학습자로 하여금 쉽게 학습을 할 수 있게 하였다. 앞으로 본 연구는 정렬 알고리듬뿐만 아니라 더욱 다양한 알고리듬에 대한 학습 애플릿을 개발하여 학생들이 더욱 다양한 학습을 할 수 있게 하여야 할 것이다.

[참 고 문 헌]

[1] Allen Holub, Taming Java Threads, Apress, 2001

[2] Gary Cornell, Cay S. Horstmann, core JAVA , SunSoft Press, 1997

[3] Scott Oaks & Henry Wong, Java Threads, 2nd Edition , O'REILLY, 1999

[4] 이현우, 천영환, Java Programming Bible Ver.2, (주)영진출판사, 2000

[5] 이재규, C로 배우는 알고리즘 , 世和, 1994

[6] 강맹규, 자료구조 , 홍릉과학출판사, 1995

[7] 최관순, 최규오, 전홍구, 웹을 기반으로하는 C 프로그램 작성 및 실행에 관한 연구 , 한국신호처리시스템학회 논문집, 2001