1. 서론
기존의 정렬 알고리즘의 성능을 향상시키기 위하여 제안된 정보블록 정렬 알고리즘(IBPA: Information Block Preprocessing Algorithm)은 데이터의 분포 상태를 파악한 정보에 따라 정보블록을 생성하고, 이 정보를 이용하여 데이터를 재배치하여, 각각의 복록을 기존의 정렬 알고리즘을 적용하여 최종 정렬을 수행함으로써 정렬 효과를 향상시키는 정렬 알고리즘을 말한다[1]. 본 논문에서는 IBPA를 이용한 새로운 정보블록 정렬 알고리즘 즉, 정보블록 알고리즘(IBSA: Information Block Sort Algorithm)을 제안하였다.

IBSA는 두 가지로 나누어지는데, 하나는 일반적인 데이터를 처리하는 경우의 정렬 알고리즘(IBSA_a)이며, 나머지 하나는 복록이 없는 데이터를 처리하는 정렬 알고리즘(IBSA_b)이다. 2장에서는 이용된 용어의 정의, IBSA에 필요한 메모리 및 자료 구조 그리고 알고리즘을 서술하였다. 3장에서는 IBSA의 성능을 실험에 의하여 제시하였으며, 4장에서는 결론을 나타내었다.

2. 정보블록 정렬 알고리즘
2.1 용어의 정의
다음은 본 연구에서 사용된 용어에 대한 정의와 그것에 대한 설명이다.
1) DB(Data Block) : 데이터 블록. 리스트를 M개로 나누었을 때 나누어진 각 부분을 말하며, 정보블록은 다르지만 개수는 같음.
2) IB(Information Block) : 정보블록.
3) BDR(Block Data Range) : 하나의 데이터 블록에 포함될 수 있는 데이터의 범위를 수치화한 값 (1≤BDR).
4) M : 정보블록의 개수 (M≤N).
5) 복록일 : 하나의 데이터가 임시기억장소로 이동되어 비어있기 시작한 때부터 다시 이곳에 다른 데이터가 들어올 때까지의 작업기간을 의미한다.

A New Sort Algorithm : Information Block Sort Algorithm(IBSA)
Tae-Ok Song Tae-Young Kim
Dept. of Computer Education, Korea National University of Education

요 약
본 논문에서는 정보블록 알고리즘(IBPA: Information Block Preprocessing Algorithm)을 이용한 정보블록 정렬 알고리즘(IBSA: Information Block Sort Algorithm)을 제안하고 그 성능을 평가하였다. IBSA의 시간복잡도는 O(N)이며, 데이터의 분포 상태에 영향을 받지 않는다. IBPA의 성능을 측정해본 결과, 2픽셀간의 데이터를 정렬한 경우, 중복값 없는 경우(a)는 백 정렬의 32.42%, 기수정렬의 9.9% 및 비교화수만으로도 정렬할 수 있음을 보여주었으며, 중복값 없는 경우(b)는 백 정렬의 53.12%, 기수정렬의 12.79% 및 비교화수만으로도 정렬할 수 있음을 보여주었다.

2.2 자료구조 및 메모리 사용량
IBSA에서 이용되는 데이터에는 DataInfo와 CopyData라는 두 가지 자료 구조가 있다. CopyData는 데이터블록에 있는 데이터를 그대로 복사한 것으로서, 자료구조는 데이터블록과 동일하다.

DataInfo는 각각의 데이터 블록을 정렬하기 위하여 필요한 자료구조로서, 데이터블록에 대한 정보를 저장하고 있다.

DataInfo의 구성요소는 <표 1>과 같다.

<table>
<thead>
<tr>
<th>구성요소</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value(V)</td>
<td>데이터 값</td>
</tr>
<tr>
<td>Data_Count(DC)</td>
<td>Value 값의 개수</td>
</tr>
<tr>
<td>Accumulated(Position(AP))</td>
<td>누적 시작 위치. CopyData에서 이동되어야 할 메모리의 기준위치값을 의미한다.</td>
</tr>
</tbody>
</table>
| Sent_Count(SC) | CopyData에서 데이터블록으로 데이터가 정렬된 값('Value' 값의 데이터 개수)

데이터를 정렬하기 위해서 필요한 메모리로 IBPA에 필요한 메모리에 DataInfo에 할당된 메모리와 CopyData에 할당된 메모리를 더한 값이다.

필요한 메모리 = 리스트의 크기 + 정보블록의 크기 + DataInfo의 크기 + CopyData의 크기
정보블록의 크기 = M + 단위정보블록의 크기
단위정보블록의 크기 = 6 x sizeof(요소의 데이터형)

2.3 알고리즘
IBSA를 이용하는 IBSA의 알고리즘은 (그림 1)과 같다.

Part 1. IBPA 정처리
Part 2. 정렬 : M만큼 반복
Step 1. TDataInfo 메모리 할당 및 초기화 (누적 시작 위치 계산)
Step 2. TCopyData 메모리 할당 및 복사
Step 3. 원본 Data 배열의 데이터 이동
Step 4. 동적 메모리 할당 해제

(그림 1) 정보블록 정렬 알고리즘

* 한국교원대학교 컴퓨터교육과 박사과정
* 한국교원대학교 컴퓨터교육과 교수
** 이 논문은 2000년도 무한한학 21 사업 핵심분야에 의해 지원되었음
2.5 시간복잡도
IBSA의 경우, 데이터가 하나의 데이터 블록에 집중되는 경우와 같은 특약의 경우는 효과가 없다. 하지만 IBSA의 경우, IBBA는 다른 데이터 블록에 있는 데이터의 개수가 많을수록 데이터의 사용량은 증가하지만 시간복잡도는 극단한 영향을 미치지 않는다.

Part.1의 비교를는 IBSA의 비교와 동일하다. 그러므로 Part.1의 비교는 아래와 같이 시간복잡도는 \( O(N) \)과 같다.

\[
\text{Part.1의 비교} \equiv \Theta(N)
\]

Part.2의 비교는 아래와 같이 나타낼 수 있으며, 시간복잡도는 \( O(N) \)과 같다.

\[
\text{Part.2의 비교} \equiv \Theta(N)
\]

그러므로 Part.1과 Part.2의 비교를 더하여 IBSA의 시간복잡도를 나타내면 <표 2>과 같이 나타낼 수 있다.

<table>
<thead>
<tr>
<th>표 2</th>
<th>IB 정렬 알고리즘의 시간복잡도</th>
</tr>
</thead>
<tbody>
<tr>
<td>시간복잡도</td>
<td>( T(N)=O(N) )</td>
</tr>
</tbody>
</table>

중복값이 없는 데이터를 정렬하는 경우, IBSA는 종복값이 있는 데이터를 정렬하는 경우보다 더욱 간단한 알고리즘을 통해 결과적으로 시간복잡도가 낮아지게 된다.

3. IB 정렬 알고리즘의 성능 평가
3.1 실험 조건
IBSA의 성능을 측정하기 위한 실험조건은 다음과 같다.

1) 정렬알고리즘: 최적정렬과 기수정렬을 실험대상으로 선정하였다. 기수정렬은 기수교환정렬(radix exchange sort)로서, 32비트의 기(key)가 사용되었다. 그리고 중복값이 없는 경우도 실험 알고리즘에 포함시켰다.
2) 데이터의 종류: 비교적 고른 분포를 보이는 난수발생기(random generator)로 생성된 랜덤 데이터로 실험이야.
3) 데이터의 개수: BDR가 50으로 설정된 상태에서 1,000개를 측정단위로 하여 2백만개 데이터까지 실험이야.
4) 비교수: 'if'와 'while' 그리고 'for'와 같은 명령문에서 도 실제로는 비교가 발생하므로 정확한 비교수를 측정하기 위해 이러한 명령문 역시 비교수에 누적되었다.

3.2 실험 결과
정렬알고리즘의 실험결과를 <표 3>과 (그림2)에 나타내었다.
<표 3> IBPA의 성능 측정-1

<table>
<thead>
<tr>
<th>IBPA의 성능 측정-1</th>
<th>IBPA의 성능 측정-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBPA의 성능 측정-3</td>
<td>IBPA의 성능 측정-4</td>
</tr>
</tbody>
</table>

로 산정된 시간복잡도가 실제적인 수행 시간을 의미하는 것으로 간주할 수 있다.

4. 결론
IBPA를 이용한 새로운 정렬 알고리즘인 IBBA의 알고리즘과 성능을 분석해보았다.
IBBA는 알고리즘의 빠른 계산과 정렬하는 알고리즘이며, IBBA는 성능의 잠재적 가능성을 보다 빠르다는 점에서 결정적인 평가를 받을 수 있지만, IBBA와 IBBA의 분포상태에 따라 성능이 약간 불안정하거나 문제점을 보완한다는 점에서 좀 더 나은 평가를 받을 수 있을 것이다.


참고문헌