

(Red-Black Tree)의 기본 속성과 그 왜곡 방지 방법

practice 자료구조 · 2024. 10. 14. 14:39

Q. (Red-Black Tree)의 기본 속성과 그 왜곡 방지 방법은 어떤건가요

레드-블랙 트리(Red-Black Tree)는 이진 탐색 트리의 변형으로, 트리가 **자기 균형**을 유지하기 위한 특정 규칙을 따릅니다. 이러한 규칙들이 **왜곡**(불균형)을 방지하는 역할을 합니다. 트리가 한쪽으로 치우치지 않도록 균형을 유지하기 위해 색상 변경과 회전 같은 연산을 사용합니다.

레드-블랙 트리의 기본 속성 (규칙)

- 노드는 빨간색 또는 검은색이다.**
 - 각 노드는 빨간색(Red) 또는 검은색(Black)의 색상을 가집니다.
- 루트 노드는 항상 검은색이다.**
 - 트리의 최상위 노드(루트)는 항상 검은색입니다.
- 모든 리프 노드(NIL 노드)는 검은색이다.**
 - 리프 노드(트리 말단의 NULL 노드)는 항상 검은색입니다.
- 빨간색 노드의 자식은 항상 검은색이다.**
 - 즉, 빨간색 노드가 연속으로 나올 수 없습니다. 레드 노드가 있으면 그 자식 노드는 반드시 블랙입니다. 이를 통해 트리의 균형을 유지합니다.
- 어느 노드에서든 해당 노드로부터 그 자손 리프 노드에 도달하는 경로에서 만나는 모든 검**

은색 노드의 수는 같다**.

- 이 규칙은 트리의 경로 길이를 일관되게 유지함으로써 균형을 유지합니다. 검은색 노드의 수가 동일해야 하기 때문에, 어떤 경로가 다른 경로보다 상대적으로 더 길어질 수 없습니다.

레드-블랙 트리의 **왜곡 방지 방법**

레드-블랙 트리는 삽입과 삭제 시 균형이 깨질 가능성이 있습니다. 이를 해결하기 위해 두 가지 주요 방법을 사용합니다: **색상 변경**과 **회전(rotate)**. 이 두 가지 방법을 통해 트리의 균형을 유지하여 높이가 $O(\log n)$ 에 머물도록 합니다.

1. **삽입 시 왜곡 방지 방법**

레드-블랙 트리에 새로운 노드를 삽입할 때 트리의 균형이 깨질 수 있습니다. 이때 트리가 왜곡되지 않도록 하기 위해 다음과 같은 조정을 합니다:

- **색상 변경 (Recoloring)**: 삽입된 노드와 그 부모, 조부모의 색을 재조정하여 규칙을 만족하도록 합니다. 특히 **연속적인 레드 노드가 발생하지 않도록** 색상을 바꾸는 경우가 많습니다.
- **회전 (Rotation)**: 만약 색상 변경만으로 트리 균형을 맞출 수 없다면, **왼쪽 회전** (left rotation) 또는 **오른쪽 회전** (right rotation)을 사용하여 트리 구조를 조정합니다. 회전은 트리의 한쪽이 과도하게 깊어지는 문제를 해결하여 균형을 유지합니다.

- **왼쪽 회전**: 부모 노드를 기준으로 서브트리 전체를 왼쪽으로 회전시킵니다.
- **오른쪽 회전**: 부모 노드를 기준으로 서브트리 전체를 오른쪽으로 회전시킵니다.

이 회전 과정을 통해 트리의 불균형을 방지하고, 각 서브트리가 균형을 이루도록 조정합니다.

2. **삭제 시 왜곡 방지 방법**

레드-블랙 트리에서 노드를 삭제할 때도 트리의 균형이 깨질 수 있습니다. 이 경우, 트리의 균형을 복구하기 위해 아래의 방법을 사용합니다:

- **색상 변경 (Recoloring)**: 삭제 후 남은 노드들 사이에서 검은색 노드의 수가 일치하지 않으면 색을 변경해 블랙 밸런스를 맞춥니다.
- **회전 (Rotation)**: 만약 삭제로 인해 트리가 불균형해지면 **왼쪽 또는 오른쪽 회전**을 통해 균형을 맞춥니다.

3. **회전과 색상 변경의 예시**

- ****삽입 후 회전 예시****: 새로운 노드를 삽입했을 때, 만약 그 부모가 빨간색이라면 연속된 빨간색 노드 규칙을 위반하게 됩니다. 이때 색상 변경을 통해 부모를 검은색으로 바꾸거나, 삼촌 노드와 함께 색을 조정할 수 있습니다. 또한, 트리의 구조가 크게 기울어졌을 경우 회전을 통해 균형을 맞춥니다.

- ****삭제 후 회전 예시****: 노드를 삭제한 후에도 트리의 균형이 깨질 수 있습니다. 이 경우도 색상 변경과 회전을 통해 트리의 깊이를 조정해 균형을 유지합니다.

요약

****레드-블랙 트리****는 삽입과 삭제 시 트리의 ****균형을 자동으로 유지****하는 자료구조입니다. ****5가지 기본 규칙****을 통해 트리의 균형이 유지되며, 균형이 깨질 때는 ****색상 변경****과 ****회전****을 통해 왜곡을 방지합니다. 이러한 방법을 통해 레드-블랙 트리는 모든 연산에서 ****시간 복잡도를 $\mathcal{O}(\log n)$ ****으로 보장합니다.

♡ 공감



000

구독하기

'practice' 자료구조 카테고리의 다른 글

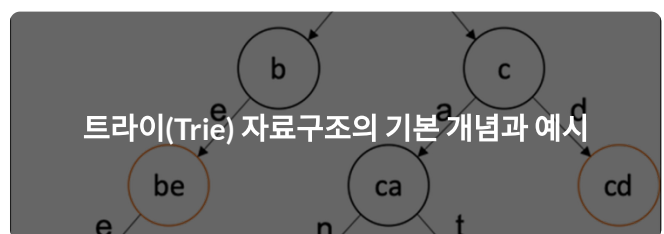
B- / B+ 트리 차이점 (0)	2024.10.14
트라이(Trie) 자료구조의 기본 개념과 예시 (0)	2024.10.14
사이클 탐지 (0)	2024.10.14
그래프 탐색- DFS, BFS (0)	2024.10.14
P / NP (0)	2024.09.30

관련글

[관련글 더보기](#)

데이터 저장	모든 노드(내부 노드와 리프 노드)가 데이터를 저장함	데이터는 오직 리프 노드에만 저장됨
내부 노드	데이터와 포인터가 함께 저장됨	포인터만 저장되고, 데이터는 리프 노드에 저장됨
탐색 경로	데이터는 어디서든 찾을 수 있음	리프 노드에서만 찾을 수 있음
리프 노드	리프 노드가 서로 연결되지 않음	리프 노드들이 Linked List로 연결되어 있음
순차 접근 성능	리프 노드 간에 직접 연결이 없으므로 순차 접근 성능이 낮음	리프 노드들이 연결되어 있어 순차 접근이 더 빠름

B- / B+ 트리 차이점



트라이(Trie) 자료구조의 기본 개념과 예시

DFS vs BFS 비교

사이클 탐지

탐색	DFS	BFS
탐색 방식	깊이 우선 (먼저 깊이 탐색)	너비 우선 (먼저 넓이 탐색)
자료 구조	스택(Stack) or 재귀(Recursion)	큐(Queue)
적용 상황	경로 탐색, 사다리 타기	단거리 탐색
시간 복잡도	$O(V + E)$	$O(V + E)$
공간 복잡도	$O(V)$ (최악의 경우)	$O(V)$
주요 특징	경로의 깊이를 먼저 탐색, 막히면 백트래킹	최단 경로 보장, 가까운 노드부터 탐색

그래프 탐색- DFS, BFS

자연어(NLP)

네이쳐2024 님의 블로그입니다.

구독하기 +

댓글 0



이름

비밀번호

내용을 입력하세요.



등록