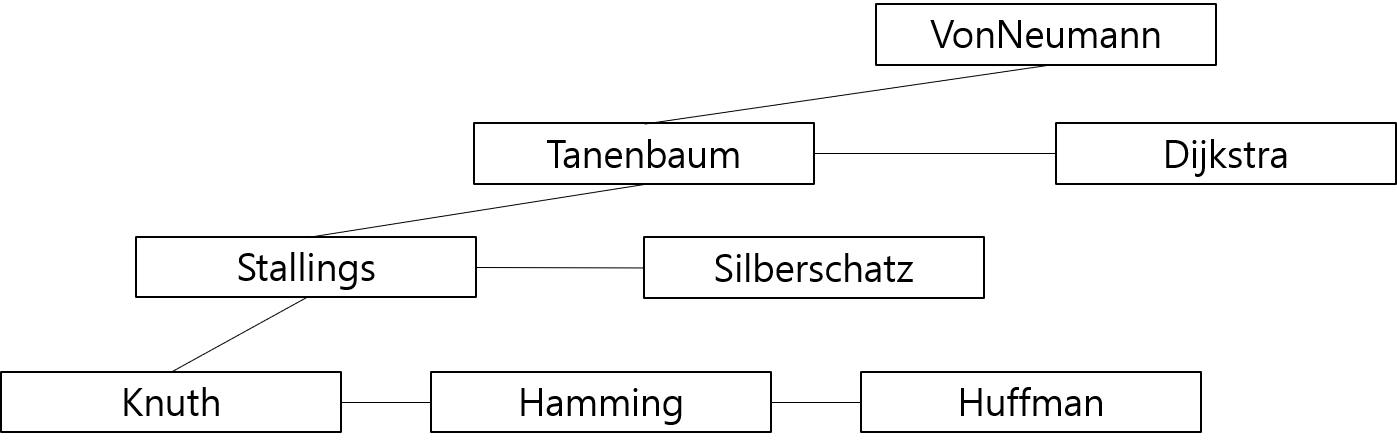
자료구조와실습  
- Project #3 -

동국대학교 컴퓨터공학과  
2016112122 이한빈



문제해결을 위해 일반적인 Binary Tree 구조를 응용하여 사용한다. 위는 주어진 tree를 left-child, right-sibling 구조로 변환한 모습이다. 위 구조를 토대로 프로그램을 구현하려 한다.

|  |
| --- |
| template < class DT, class KF >  class BSTreeNode // Facilitator for the BSTree class  {  private:  // Constructor  BSTreeNode ( const DT &rootData,  BSTreeNode \*leftPtr, BSTreeNode \*rightPtr );  // Data members  DT dataItem; // Binary search tree data item  BSTreeNode \*left, // Pointer to the left child  \*right; // Pointer to the right child  friend class BSTree<DT,KF>;  }; |

Lab07: Binary Search Tree에서 사용한 Node이다. pointer 변수 left는 left child에 대한 link, right는 right child에 대한 link를 갖는다. 이 문제에서는 자식 노드가 여러 개가 올 수 있기 때문에 그만큼 자식 노드에 대한 레퍼런스가 늘어나게 된다.

|  |  |
| --- | --- |
| dataItem | |
| left – child | right - sibling |

degree가 끝없이 늘어날 수 있는 Tree 구조에서는 각 자식에 대한 레퍼런스 수를 무한정 늘려 저장하는 것은 비효율적이다. 따라서, 나는 프로그램을 구현할 때 Tree Node의 left에는 자식 노드를, right에는 형제 노드에 대한 링크를 저장하고자 한다. 이러한 구조는 메모리의 낭비를 줄이고, 시간을 효율적으로 사용할 수 있을 것이다.

Result:

