• ( 6 Points) Problem 12.12
• ( 6 Points) Problem 12.15
• Additional Problems
  - A8.1 [6 points] Calculate the maximum number of nodes, search-key values, and pointers of a $B^+$-tree at each of the following levels of indexing, assuming that each (leaf/non-leaf) node allows up to 25 pointers, and hence 24 search-key values.

<table>
<thead>
<tr>
<th>Level</th>
<th>No. of nodes</th>
<th>No. of search-key values</th>
<th>No. of pointers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  - A8.2 Given the following relational database:

    \[\text{Student} \ (\text{Name, SSN, GPA, Major}), \ \text{Key: SSN}\]
    \[\text{Has\_taken} \ (\text{Course\#, SSN, Semester}), \ \text{Key: Course\#, SSN}\]

Assume that the following properties hold:
* $\text{Student}$ has 10,000 tuples, and 10 tuples of $\text{Student}$ fit in one data block.
* $\text{Has\_taken}$ has 4,000 tuples, and 20 tuples of $\text{Has\_taken}$ fit in one data block.
* $V(\text{Name, Student}) = 250$ and $V(\text{Major, Student}) = 40$, where $V(A, r)$ stands for the number of unique values of attribute $A$ in relation $r$.
* A primary, $B^+$-tree index for $\text{Name}$; a secondary, $B^+$-tree index for $\text{Major}$.
* 50 pointers fit in one index block.

1. [4 points] Estimate the number of tuples in $\text{Student} \bowtie \text{Has\_taken}$.

2. Consider the following expression $E$ in SQL:

    \[\text{SELECT SSN FROM Student WHERE Major = 'CS' AND Name = 'Adams'.}\]

Using each of the following indexing strategies, estimate the cost of access, i.e., number of block accesses, for processing the expression $E$ above (justify your answer for full credit):
(a) [6 points] Use the primary index on Name.
(b) [6 points] Use the secondary index on Major.

3. Using each of the following join strategies, estimate the number of block accesses required for $\text{Student} \bowtie \text{Has\_taken}$.
(a) [4 points] Nested-loop join
(b) [4 points] Block nested-loop join
(c) [4 points] Merge join
(d) [4 points] Use indices, assuming that there is a $B^+$-tree index for SSN in $\text{Has\_taken}$, $V(\text{SSN, Has\_taken}) = 1,000$, and there is no indexing defined on any attribute in $\text{Student}$.