1. (7 points) Let’s turn yet again to our favorite university DB example:

\[
\text{Prof}(\text{pno}, \text{pname}, \text{salary}, \text{age}, \text{email}) \quad -- \text{info about Professors}
\]

\[
\text{In}(\text{pno}, \text{dno}, \text{percent}) \quad -- \text{Professors can be In several departments}
\]

\[
\text{Dept}(\text{dno}, \text{dname}, \text{college_id}, \text{chair_pno}) \quad -- \text{info about Departments}
\]

Consider the B+ tree index on Prof.age pictured below:

(a) (2 pts) Draw what this index will look like after firing the 35-year old professor(s):

(b) (2 pts) How many page reads will the delete operation take? How many page writes?

\[
\text{# reads: } 4 \quad \text{# writes: } 3
\]

(c) (1 pt) Suppose a frequent query asks for professors in a specified age range. Ideally, should this index be clustered or unclustered? (x) Clustered ( ) Unclustered
(d) (2 pts) Draw what the original B+ Tree index from the previous page will look like after hiring a new 42-year old professor:

(e) (8 pts) Write a SQL query that this B+ tree index enables an index-only plan for:

```
SELECT age FROM Prof;
SELECT DISTINCT(age) ...
SELECT MIN(age) ...
SELECT MAX(age) ...
SELECT AVG(age) ...
```

(2) (2 points) Consider next the static hashed index on Prof.pno pictured below:

(a) (1 pt) How many page reads will it take to locate the professor whose professor number is 36? # reads: __1____

(b) (1 pt) Draw what the static hashed index will look like after hiring a new professor with a professor number of 76.