Quiz 6 Answers, Fall 2018
CSCI 4380 Database Systems

```
Candidates(personid, eid, year, partyname, ballotposition, numvotes)
Elections(id, name, etype, state, howoften)
```

Question (Points: a:8, b:8, c:9). You are given the following statistics:

```
TUPLES(Candidates) = 8000, PAGES(Candidates) = 500
TUPLES(Elections) = 1200, PAGES(Elections) = 120
TUPLES(Candidates.partyname = 'Morty Party' and Candidates.year = 2018) = 40
```

Index name	Indexed Attributes	Structure	# pages at leaf level
CanIdx1	Candidates(eid, year, partyname)	3 levels (root, internal, leaf)	80 pages

The following are the queries over this table.

```
Q1: select * from candidates c, elections e
    where c.partyname = 'Morty Party' and c.year = 2018 and c.eid=e.id;
```

Answer the following with explanations of your computation.

(a) What is the cost of sorting `Candidates` given **M=10** blocks. Show your work.

Answer here.

Step 1: Read once, write one, create 50 sorted groups, cost=1,000

Step 2: Read once, write one, create 5 sorted groups, cost=1,000

Step 2: Read once, sort and output, cost=500

Total = 2,500

(b) What is the cost of the following query `Candidates` ⋈ `Elections` using block-nested loop join with **M=11** blocks. Show your work.

Answer here.

500 + 120*500/10 = 6,500

We will also accept: 120 + 120*500/10 = 6,120

(c) What is the cost of the following query plan for `Q1`:

Search candidates on condition `c.partyname = 'Morty Party' and c.year = 2018` using `CanIdx1` with sufficient memory.

Pipeline the results to a block-nested loop join with elections with `M=11` blocks. (Hint: The number of tuples satisfying partyname and year condition is given. How many pages does this take to keep in memory?)

Answer here. Scan the whole leaf level (as it is not sorted first by the queried attributes)

Index cost: 2+ 80 + 40 (to read other attributes) = 122

The result fits in 40/ (8000/500)= 3 pages. Since the outer relation fits in memory completely, we can complete the join by reading Elections only once.

Total cost = 122 (for index search) + 120 (for join) = 242 pages.