CPSC 404 Advanced Relational Databases Fall 2003.
Quiz 2 Solutions.

QUESTION 1.

# pages in Employee = 2 x 10^6 x 100 / 4000 = 50,000.

partition phase: Use a hash function to create 500-1 = 499 partitions of Employee. We need 1 page for the input buffer. Now, we know tuples in a partition can only have duplicates in the same partition. The cost of this step is reading each page and writing it to disk, so it would cost 2 x 50,000 = 100,000 I/Os.

"merge" (i.e., duplicate elimination) phase: read each partition into buffer, one by one. Use a second hash function to build an in-memory hash table. Then eliminate duplicates. Assuming no significant skew in the data, each partition is on an average 50,000/499 = 100 pages (approx).
So, we have enough room to read in each partition in full into buffer, so there is no need for recursive hash partitioning. There is no writing charge so the only cost of this step is reading in the 50,000 pages of Employee into buffer again. (This is a slight overestimate as we could have eliminated some, but not all, duplicates in the partition phase.)

Total cost = 100K + 50K = 150,000 I/Os. Note that for duplicate elimination, we need to hash on all attributes of the relation.

QUESTION 2.

(a) Assuming uniform distribution of title-values, the 8 distinct title values are equally likely. So, the reduction factor of title > 3 = 5/8.
//Note that the # tuples in Employee plays no role here! It's only needed if we need to know how many tuples satisfy title > 3. Even if you used the size of Employee (i.e., 2.1 million tuples in this question) and divided it up evenly among the 8 title values and estimated the reduction factor based on that, you'd still get RF = 5/8.//

(b) Equi-width histogram:

bucket	hits
1-2	550K
3-4	400K
5-6	900K
7-8	250K

RF of title > 3 can be calculated as follows:

Contribution from bucket 3-4 = ½ x 400K = 200K. This is because one out of the two values in this bucket(3 and 4) satisfies title > 3.
Contribution from full buckets 5-6 and 7-8 = 900+250K = 1150K.

total estimated # tuples satisfying title > 3 = 200 + 1150 K = 1350K.
RF = estimated # tuples satisfying condition / total # tuples
   = 1350K/2100K = 9/14 = 0.64 (approx).

(c) Equi-depth histogram:

bucket	hits
1	500K
2-4	450K
5	550K
6-8	600K

RF of title > 3 can be calculated as follows:

contribution from bucket 2-4 = 1/3 x 450K = 150K.
contribution from full buckets 5 and 6-8 = 550 + 600 K = 1150K.
total estimated # tuples satisfying title > 3 = 150 + 1150 K = 1300K.
RF = estimated # tuples satisfying condition / total # tuples
   = 1300K/2100K = 13/21 = 0.62 (approx).

QUESTION 3.

Emp is 200,000/20 = 10,000 pages.
Dept is 40,000/20 = 2,000 pages.
Available buffer size = 500 pages.

(a) Reserve 1 page for inner relation, 1 page for the output buffer, and the remaining 500 - 2 = 498 pages for a chunk of the outer relation. We should put Dept. in the outer relation, since this would minimize the total I/O cost. Then the # chunks = ceil(2000/498) = 5. For each chunk of Dept. in the buffer, read Emp in page by page, join and write out the result. If we don't charge for output writing, the total cost is 2000 (for reading Dept) + 5 x 10000 (for reading Emp 5 times) = 52,000 I/Os.

(If we put Emp in the outer loop, we would have ceil(10000/498) = 21 chunks. The total cost would then be 10000 + 21 x 2000 = 52,000 I/Os. So, the estimated cost is the same for this choice, in *this* problem.)

(b) The minimum number of I/Os required by sort-merge join is 3 x (10000 + 2000) = 36000.
The min. # buffer pages needed to make this happen is sqrt(10000) = 100.
But we have 500 (which is > 100) buffer pages so we are okay.