Quiz Date: 21.11.2011

CS351 – FALL '11 – QUIZ #3 – Sec01

Question (a and b): Consider linear hashing . What is *bv, h, # of pages at h, and # of pages at (h+1)* when number of pages *n* is (a) 6, (b) 21 ?

Answer (a and b):

$h = \lfloor \log_2 n \rfloor$

$bv = n - 2^h$

$\#\text{ of pages at } h = 2^h - bv$

$\#\text{ of pages at } (h+1) = 2*bv \text{ } or \text{ } n - (\#\text{ of pages at } h)$

a) $h = 2, bv = 2, \#\text{ of pages at } h = 2, \#\text{ of pages at } (h+1) = 4$

b) $h = 4, bv = 5, \#\text{ of pages at } h = 11, \#\text{ of pages at } (h+1) = 10$

Question (c): Consider linear hashing with h = 1, bv = 0, Desired Lf = 2/3, and Bkfr = 3. Insert the following records: 27, 19, 64, 12, 33.

Answer (c): Note that we use suffixes!

Decimal=> Binary, 27=> 11011, 19=> 10011, 64=> 1000000, 12=> 1100, 33=>100001

			0
27			1

			0
27	19		1

64			0
27	19		1

64	12		0
27	19		1

At this point the desired load factor level has been reached to have a file update we need to add 2 more records.

64	12		0
27	19	33	1

Note that in this example we may even proceed without converting the numbers to binary. Just pay attention if they are even (1-suffix=0) or odd (1-suffix=1).

Quiz Date: 25.11.2011

CS351 – FALL '11 – QUIZ #3 – Sec02

Question: Consider linear hashing.. What is *bv*, *h, # of pages at h, and # of pages at (h+1)* when number of pages *n* is (a) 6, (b) 23, (c) 278 ?

Answer:

$h = \lfloor \log_2 n \rfloor$

$bv = n - 2^h$

$\#\text{ of pages at } h = 2^h - bv$

$\#\text{ of pages at } (h+1) = 2*bv \textit{ or } n - (\#\text{ of pages at } h)$

a) h = 2, bv = 2, # of pages at $h = 2$, # of pages at $(h+1) = 4$

b) h = 4, bv = 7, # of pages at $h = 9$, # of pages at $(h+1) = 14$

c) h = 8, bv = 22, # of pages at $h = 234$, # of pages at $(h+1) = 44$

d) Quiz Date: 26.10.2011

e) CS351 – FALL '11 – QUIZ #3 – Sec03

Question: Given *n* = 200,000, *R* = 400 bytes, *B* = 2400 bytes, *ebt* = 0.5 ms
What is the time needed to generate the sorted segments using heap sort?

Answer:

Time to read unsorted file = b * ebt = (200,000 * 400 / 2400) * 0.5 ~= 16,7 s
Time to write unsorted file = b * ebt = 16,7 s
Total time ~= 16,7 + 16,7 ~= 33,4 s [time for (s+r)'s are neglected]

Optional Question: Consider the above scenario.. Given 10 MB in main memory, how many (s+r)'s does sorting take?

Answer:

File size = 200,000 records * 400 bytes = 80 MB
80 / 10 = 8 MB for each possible r/w
Then 8*(s+r) for reading + 8*(s+r) for writing = 16*(s+r)