23

Fall 2013, CS288 Test 1, 1-2:15 pm, Fri, 9/27/2013, CKB204 61

Name: Toufic Hamadeh

Make sure you have four pages. Do not take any page(s) with you. Any missing page(s) will result in failure in the exam. This exam is closed book close notes. Do not exchange anything during the exam. You all have the same exam. **No questions will be answered during the exam, including typos.** I don't want to give different answers to different people. If you are in doubt, briefly state your assumptions below, including typos if any.

I have read and understood all of the instructions above. On my honor, I pledge that I have not violated the provisions of the NJIT Academic Honor Code.

Signature: Toufic Hamadeh **Date:** 9/27/2013

Questions 1-10: 5 points each

1. Write the output when you exeucte the Bash command: `echo \"Hello  World\"` Note there are *three* space characters between `Hello` and `World`. Indicate the number of space charecters in the output after you execute the command.

"Hello␣␣World" 1 space

2. Given `x="hello"`, executing the Bash command `y=\`expr $x * 2\`` will result in
(a) hello2 (b)hellohello (c) command not found (d) errors (e)none of the above

3. Write a Bash command to change the permission of a file `f` to allow for the user to read/write/execute, for the group members read/execute, and the others read only. Use *octal* numbers only as discussed in class. Use of any other format will receive no credit.

chmod 755 f → 754

4. Given `lst=(1 2 3)`, write a Bash command to find the number of elements in `lst`.

echo "${#lst[@]}"

5. Given `l1=(1 2)` and `l2=(x y)`, write a Bash command to obatin `lst=(x y 1 2)`

lst = ($l1 $l2) lst = (${l2[@]}␣${l1[@]})

6. Given `f()` and `main()`, what is the output when `main` is called?

```
function f() { local p=$2; q=1; r=$1; }
function main() { p=1; q=2; r=3; f $p $q $r; echo $p $q $r; }
```

output => 211 → 111

7. Given `x=abc`, the Bash command `[[ $x > 123 ]] && echo yes || echo no` will output:
(a)yes (b)no (c)command not found (d)errors (e)none of the above

8. Given `s='<span class="viewcount">/acct/1,696,807/name/</span>'`, write a Bash command to extract the numbers without commas. Your command must work for any number of numbers and commas. Answers based on fixed nubmer of numbers and commas will receive no credit.

grep \([0-9,]\)

grep [0-9,]* \s //

9. Write a grep statement to match lines such as below using two backreferences.
`Mr abc came home to Mrs abc and visit Mr xyz and Mrs xyz to discuss backreferences`

grep Mrs? \s ([a-zA-Z]*) \s.* \1

grep \([a-z]\)

10. Assuming the file `fruits.txt` shown right has eight lines of fruit names, write the output of the Bash grep command: `grep -v ^p fruits.txt`

apple –
orange
pear –
peach –
grape –
banana
blueberry
plum –

^p starting with p

apple
pear
peach
grape
plum

apple
orange
grape
banana
blue berry

11. (10 points) Write a Bash script using `for` loop to find the sum of `lst=(1 2 3 4 5)`.

```
#!/bin/bash
lst=(1 2 3 4 5)
i=${#lst[@]} = 5
let k=(i-1)   k = 4
for (( idx=0 ; idx<=k ; idx++)) ; do
    let sum=(sum + ${lst[idx]})
done
echo "$sum"
```

12. (20 points) Write a Bash script, reverse.sh, which reverses the contents of a directory passed as a parameter. Assuming `/my/dir` contains "cache cron games lib log run tmp," your program "`reverse.sh /my/dir`" will return "tmp run log lib games cron cache." Use an array and two functions: `main()` and `reverse()`. Reverse manually manipulating the list. DO NOT use the built-in command sort -r.

reverse.sh

```
#!/bin/bash

reverse()
{
idx=$k
for (( idx ; idx>=0 ; idx-- )) ; do
    echo "${List[idx]}"
done
}
main()
{
List=($1*)
i=${#List[@]}
let k=(i-1)
reverse $k
}
main $1
```

20/

$ reverse.sh /Documents

13. (20 points): Write a Bash script to traverse a directory tree in *depth*-first order using *recursion*. A seed directory is passed as a command line parameter. Return the list of *all* sub directories in complete path.

15

```
#!/bin/bash
myfunct()
{
for i in $1 ; do
    if [ -d $1/$i ]
        myfunct $1/$i
    else
        echo $1/$i
done
}
main() {
LST=($1*)
x=${LST[@]}
myfunct $x
}
main $1
```

Fall 2013, CS288 Test 2, 1-2:15 pm, Fri, 10/25/2013, CKB204

87

Name: Toufic Hamadeh

The exam assumes 32-bit Linux machines. The exam has 5 pages. Make sure you have all the pages. Do not take any page(s) with you. Any missing page(s) will result in failure in the exam. This exam is closed book close notes. Do not exchange anything during the exam. No questions will be answered during the exam. If you are in doubt, briefly state your assumptions below, including typos if any. Questions 1 to 15 are worth 3 points each.

I have read and understood all of the instructions above. On my honor, I pledge that I have not violated the provisions of the NJIT Academic Honor Code.

Signature: Toufic Hamadeh **Date:** 10/25/2013

36

Questions 1-3: Assume radix sorting of 1024 unsigned integers on a 32-bit machine with 4 passes (rounds). The integers are initially stored in lst[1024] and the sorted integers will be available in lst at the end of sorting. int buf[1024] is available as working space.

1. What is the number of buckets?

   1 << (b/p) ✓ 255

2. The bit mask in *hexadecimal* is?

   (1<<(b/p))-1 ✓ 0xFF

3. Find the number of data assignments for correcting the result after 4 passes are completed. For example, moving lst[i] to buf[j], or buf[j]=lst[i]; is a data assignment.

   2n X 0

4. For floating point radix sort, assuming exactly half (512 floats) is negative, what is the number of data assignments for correcting the result?

   2n ✓

5. Given `float f;` write a C statement to access the binary equivalent of `f`:

   *(unsigned float *)(&f) ✓

6. In class, it was demonstrated through realtime execution that selection sort is super slow, merge sort is acceptable while radix sort is fast. Explain their performance in a sentence with not more than 20 words. You may use big O notation to simplify your answer.

selection sort = $O(n^2)$ radix sort = $O(n)$
merge sort = $O(n \log n)$

7. Given `char *str = "?a???b,,,#c";` what would `strtok(str, "?");` return?

a ~~nothing – blank; *str begins with "?"~~

8. Continuing Problem 7 write fill in the blank such that strtok() will return `??b`.

`strtok(` str / NULL `, "," );`

9. Continuing Problem 8, what would `strtok(NULL, "#,");` return?

"c"

For 10-15 on the 15-puzzle state-space search, consider open=(x,y) and succ=(p,q,r). Assuming the first one in open is always picked for search,

10. find open after merging with succ using breadth-first search strategy.

open = (x,y,p,q,r) breadth → front

11. find open after merging with succ using depth-first search strategy.

open (p,q,r,x,y) depth → end

12. find open after merging with succ using best-first search strategy.

open (p,x,q,y,r)

13. What is the branching factor for the 15-Puzzle problem?

3

14. Given `struct x { int a,b; struct x *p,*q; };` what would `sizeof(struct x)` return?

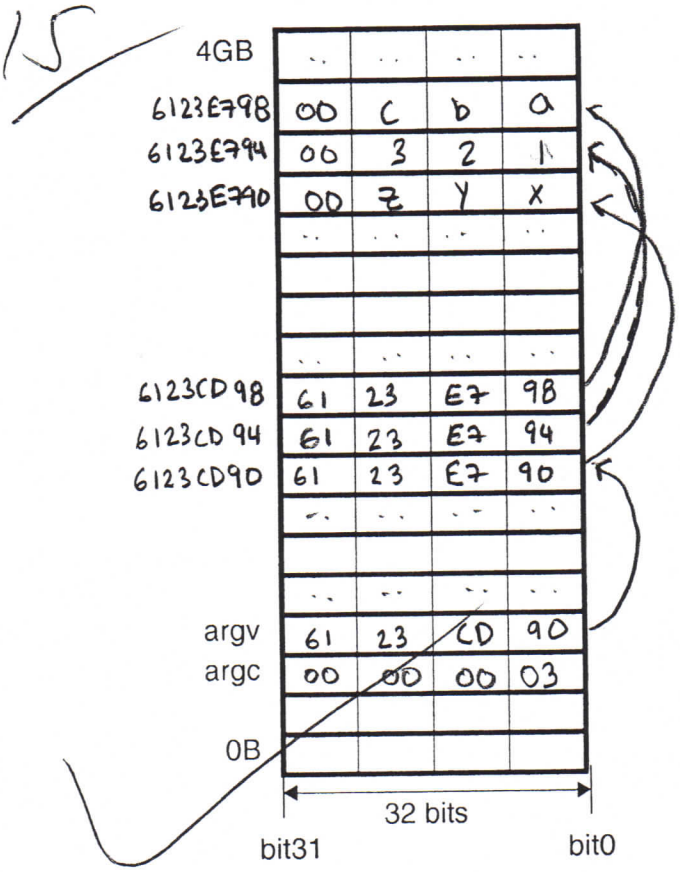
~~12 bytes~~ 16

15. Given `struct x { int **a,**b; struct x **p,**q; };` what would `sizeof(struct x)` return?

16 bytes

Problem 16 (comand line arugments - 15 points) At the command line prompt, you type "xyz 123 abc" and hit enter, where xyz is your C executable while 123 and abc are parameters. Show the contents of memory for argv, intermediate pointers, and the parameters in the table on the right. Use arrows to indicate the relationship between them.

xyz	123	abc
1	2	3



Problem 17 (Unsigned integer radix sort - 20 points): Write a C program for sorting 32-bit unsigned integers using radix sort with a group of 4 bits. Use the variables listed below. Assume lst is initialized with n numbers.

20

```
#define N 1048576
#define BIN 256
#define MAXBIT 32
#define LST 1
#define BUF 0

int n,group,bin;
int flag; /* to show which one holds numbers: lst or buf */
int lst[N],buf[N];
int count[BIN], map[BIN], tmap[BIN];

int main(int argc, char **argv){
    int i;
    flag = LST;
    initialize(); /* initialize lst with n random floats */
    for (i=0;i<MAXBIT;i=i+group) radix_sort(i); /* move lst to buf or buf to lst depending on the iteration number */
    correct(); /* sorted numbers must be in lst */
}
void radix_sort(int idx) {
    int i,j,k,mask; /* initialize mask for lifting the 4 least significant bits. */
    int *src_p,*dst_p; /* cast lst and buf to int pointers to treat lst/buf as int's */
    /* set src_p and dst_p*/
        if (flag == LST) {
            src_p = lst
            dst_p = buf
            flag = BUF }
        else { src_p = buf
               dst_p = lst
               flag = LST }
    /* count */
        for (i=0; i<bin; i++)
            { count[i] = 0
              map[i] = 0 }
    mask = 0xFF
    for (i=0; i<N; i++){
        count[(src_p[i] >> idx) & mask] ++
    }
    /* map */
        for (i = 1; i<bin; i++) {
            map[i] = map[i-1] + count[i-1]
        }
    /* move */
        for (i=0; i<N; i++){
            dest_p[map[(src_p[i] >> idx) & mask] ++] = src_p
    }
}
void correct() {
        for(i=0; i<N; i++) {
            if ((i+n)<N)
                buf[i] = lst[i+n]
            else
                buf[i] = lst[i-(N-n)]
    }
}
        for (i=0; i<N; i++)
            lst[i] = buf[i]
```

Problem 18 (Linked list - 20 points): Write two C functions append() and len() where
1. append() creates a struct clip, sets the value passed as parameter to number, and adds the newly created struct to the end of the list. *head* points to the first node in the list. Your function must be able to handle any number of nodes, including none in the beginning.
2. length() returns the number of clips in the list pointed by head.

16

```
#include <stdio.h>
void append(); int length();
struct node { int number; struct node *next; } *head;
void main() {
   struct node *head; int n;
   append(&head,1);
   n = length(head);
}

void append(struct hp , int n ) {
   struct clip *cp,*tp; /* for new clip, traversing as needed */
      tp=hp
      cp=(struct node*)malloc(sizeof(struct node));
      tp->next = cp;
      cp->number = n;
      cp->next = NULL;

      while (tp->next != NULL)
         tp=tp->next

}

int length( struct hp ) {
      int i;
      cp=(struct node*)malloc(sizeof(struct node));
      cp=hp
      while (cp->next !=NULL){
         cp=cp->next
         i++
      }
}
      return i;
```

Fall 2013, CS288 Test 3, 1-2:15 pm, Fri, 11/22/2013, CKB204 20

Name: Toufic Hamadeh 68

The exam assumes 32-bit Linux machines. The exam has 5 pages. Make sure you have all the pages. Do not take any page(s) with you. Any missing page(s) will result in failure in the exam. This exam is closed book close notes. Do not exchange anything during the exam. No questions will be answered during the exam. If you are in doubt, briefly state your assumptions below, including typos if any. Questions 1 to 17 are worth 3 points each.

I have read and understood all of the instructions above. On my honor, I pledge that I have not violated the provisions of the NJIT Academic Honor Code.

Signature: Toufic Hamadeh **Date:** 11/22/2013 27

1. The number of iterations using breadth-first search is typically smaller than that of depth-first search for the 15-Puzzle.
   (a) True (b) False (c) Cannot Determine

C 2. The depth for a state-space search tree using depth-first search is typically smaller than that of breadth-first search for the 15-Puzzle.
   (a) True (b) False (c) Cannot Determine

3. The search strategy that picks using the smallest estimated remaining distance is called
   (a) depth-first (b) breadth-first (c) branch-bound (d) best-first (e)A*

4. The search strategy that combines the distance incurred and the estimated remaining distance is called
   (a) depth-first (b) breadth-first (c) branch-bound (d) best-first (e)A*

A 5. What is the minimum number of 32-bit words to represent a 15-puzzle state? An example 15-Puzzle state is '1 2 3 4 | 5 6 7 8 | 9 10 11 12 | 13 14 15 0,' where '|' is a row delimiter.
   (a) 2 (b)4 (c)8 (d) 16 (e) None of the above

Assuming the 15-Puzzle is implemented using singly-linked lists with last node NULL terminated, answer questions 6-9.

6. Given m=number of nodes on successor, n=number of nodes on open, p=number of nodes on closed, the number of pointer assignment statements to merge a list of successor nodes with open for *depth* first search is:
   (a) O(m) (b) O(n) (c) O(p) (d) O(n)+O(m) (e) None of the above

b 7. Given m=number of nodes on successor, n=number of nodes on open, p=number of nodes on closed, what is the number of pointer assignment statements to merge a list of successor nodes with open for *breadth* first search?
(a) O(m) (b) O(n) (c) O(p) (d) O(n)+O(m) (e) None of the above

b 8. Given m=number of nodes on successor, n=number of nodes on open, p=number of nodes on closed, what is the number of pointer assignment statements to merge a list of successor nodes with open for *best* first search, assuming the open list is kept in sorted order of heuristic value?
(a) O(m) (b) O(n) (c) O(p) (d) O(n)+O(m) (e) None of the above

b 9. Given m=number of nodes on successor, n=number of nodes on open, p=number of nodes on closed, what is the number of pointer assignment statements to merge a list of successor nodes to open for *A** first search?
(a) O(m) (b) O(n) (c) O(p) (d) O(n)+O(m) (e) None of the above

10. Name an open source program that converts an html file to xhtml.

~~tagsoup~~ ~~firebug~~ X tagsoup

11. What is the Python package that provides a function to convert xhtml file to DOM tree?

~~import re~~ ~~tagsoup~~ X XML

12. List DOM node type 1 to 4.

1 = element, 2 = attribute, 3 = text, 4 = cdata

13. List the DOM node types that contain text? List numbers.

1, 2, 3

X

14. What API is required for a Python program to execute mysql queries?

import MYSQLdb ✓ connect, cursor, execute

15. The name of the method that Python uses to connect to mysql server is:

conn = MYSQLdb.connect(host=" ", user=" ", passwd=" ", db=" ")

16. The name of the method that Python uses to get access to mysql tables is:

X = conn.cursor()

17. The name of the method that Python uses to execute mysql queries is:

x.execute()
SQL command here
x.execute("""INSERT INTO Table VALUES (%s)""", (100))

Problem 18 (Search - 25 points): In HW7, you wrote a C program that performs state-space search for the 15-Puzzle similar in principle to those used in game engines.

Write a filter function which returns a list of nodes that are not on the list pointed by hp. Assume nodes_same() exists, which compares two nodes and return 0 for false or 1 for true.

```
struct node {
  int loc[N+1][N];
  struct node *next;
} *start,*goal;
int flag;
struct node *initialize(),*expand(),*merge(),*filter();
```

```
void main(int argc,char **argv) {
  struct node *open,*closed,*succ,*copen;
  open=closed=succ=NULL;
  start=initialize(argc,argv);
  open=start;
  while (open) {
    copen=open;
    open=open->next;
    succ=expand(copen);
    succ=filter(open,succ);
    succ=filter(closed,succ);
    if (goal_found(succ,goal)) break;
    if (succ) open=merge(succ,open,flag);
    copen->next=closed;
    closed=copen;
  }}
```

25

struct node *filter(struct node *hp,struct node *succ){

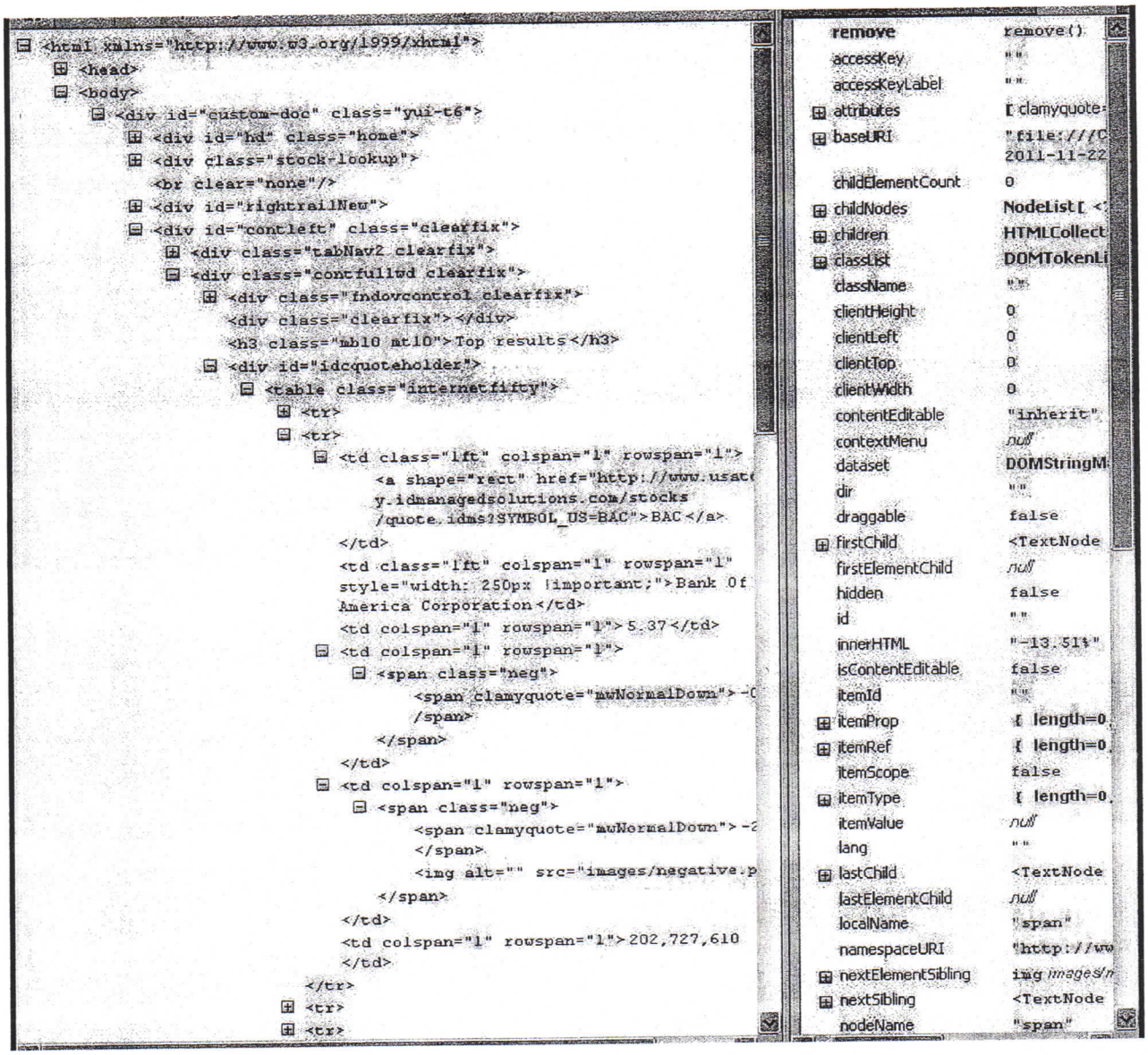
```
struct node *tp
tp = (struct node *) malloc (sizeof (struct node *)
while (hp != NULL) { tp=succ
    while (succ != NULL) {
      if ( nodes_same ( hp, succ) == 0) {
          tp = hp
          tp = tp -> next
      }
    succ = succ -> next }
hp = hp -> next }

succ = tp
```

returns nodes not ~~pointed to by~~ on list pointed to by hp

nodes_same = 0 means the nodes are not the same

```
  return succ;
}
```

Problem 19 (Linked list - 24 points): Consider the most active page in DOM displayed by Firebug. Write a Python program to return a *list of lists* each of which contains six values: symbol, name, price, change, %change, and volume: (a) write a function, get_text(), to extract the text pointed by an element pointer, (b) write a wrapper function that uses get_text().



"""

methods and properties

elm.nodeType -> returns an integer # 1=element, 2=attribute, 3=text, and 4=cdata
elm.data -> returns a string
elm.childNodes -> returns list
elm.attributes -> returns list
elm.getElementsByTagName(tag) -> returns list
elm.getAttribute(atr) -> returns a string
elm.hasAttributes() -> returns true (value) or false (None)
"""""

(a) Write a recursive Python function get_text(elm) to extract all the text of a dom tree pointed by *elm*.

12

```
lst = []
if e.nodeType in (3,4):
    if (e.data != "\n"):
        lst.append(e.data)
else:
    lst2 = e.childNodes
    for x in lst2
        lst = lst + get_text(x)
return lst
```

(b) Write a Python function to find a *list of lists* for the dom tree described in the previous page. Use get_text(elm) in part (a).

4

```
lst = []
l = get_text(elm)
for x in (len(l))
    if e.childNodes
        lst = lst + e.childNodes
return lst
```

```
src = e.getElementsByTagName("tr")
lst = []
for x in src
    lst = lst + get_text(x)
return lst
```