CS69001 Computing Laboratory – I

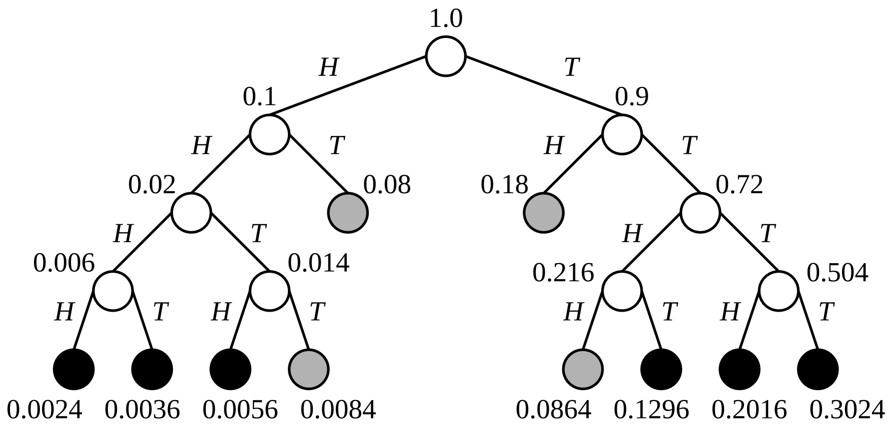
EVEN PC **Lab Test 1** **Date: 11–September–2019**

You are given n biased coins with heads occurring with probabilities $p_0, p_1, p_2, \ldots, p_{n-1}$. You keep on tossing the coins in the given order (once each), and stop as soon as one of the following two events happens.

1. You get an equal number of heads and tails.
2. You have tossed all of the n coins.

Both the events may happen together (for even n). You say that your experiment succeeds if Event 1 happens (irrespective of whether Event 2 happens together or not).

Model this random process as a binary tree. Each left-child link stands for the occurrence of a head H, and each right-child link stands for the occurrence of a tail T. Each node in the tree stores the probability of the toss outcomes so far. The following figure shows the probability tree for four coins with head probabilities $0.1, 0.2, 0.3, 0.4$. The leaf nodes represent the termination of the random process. The gray leaves stand for successful termination, and the black leaves stand for unsuccessful termination. In this example, the success probability is $0.08 + 0.18 + 0.0084 + 0.0864 = 0.3548$, and the failure probability is $0.0024 + 0.0036 + 0.0056 + 0.1296 + 0.2016 + 0.3024 = 0.6452$. The sum of these two probabilities must be 1.



Part 1: Tree data type (4)

Define a data type to store a node in the probability tree (a binary tree). Each node should consist of a floating-point (data type `double`) probability, two child pointers (left and right), optionally a parent pointer (may be helpful in Part 3), and nothing else. Only probabilities are stored in the nodes. Do *not* store the events (like TTH) anywhere in the tree. The node for any event can be reached by following appropriate pointers. Do *not* store, in any leaf node, whether this is a case of successful termination or not. In the remaining parts, you work only with the tree or the input array P.

Part 2: Build the tree (12)

Write a function *buildtree* that takes the array P of n head probabilities as input. The function recursively creates the entire probability tree, and returns a pointer to the root node of the tree.

Part 3: Tree functions (4×3)

Write functions that take as input the tree built in Part 2, and does the following (also see sample output).

allevents Print all termination events along with their respective probabilities and the information whether these are cases of successful or unsuccessful termination.

extremeevents Find the terminating events with minimum and maximum probabilities in both the cases of successful and unsuccessful termination.

successprob Compute the total probabilities of successful and unsuccessful termination.

Part 4: Success probability without the tree (8)

Write a function *notreecomp* that takes as input only the array P of n head probabilities (and not the tree built in Part 2), and computes the probabilities of successful and unsuccessful termination. The tree of Part 2 contains $\Theta(2^n)$ nodes, so the functions of Part 3 take time exponential in n. The complexity of *notreecomp* should be polynomial in n. More specifically, it should run in $O(n^2)$ time, and use only $O(n)$ extra space.

Part 5: The main function (4)

The user enters n followed by the probabilities $p_0, p_1, p_2, \ldots, p_{n-1}$. Store these probabilities in an array P of **`double`** variables. Call *buildtree* (Part 2) to build the probability tree. Call the functions of Part 3 one by one. Finally, call *notreecomp* of Part 4, and print the success and failure probabilities.

Output (10)

```
n = 6
Head probabilities: 0.219 0.213 0.457 0.767 0.291 0.503

+++ All termination events                                                          (4)
    Event:  HHHHHH   Probability = 0.002393295126    [Unsuccessful termination]
    Event:  HHHHHT   Probability = 0.002364746874    [Unsuccessful termination]
    Event:  HHHHTH   Probability = 0.005831086750    [Unsuccessful termination]
    Event:  HHHHTT   Probability = 0.005761531043    [Unsuccessful termination]
    Event:  HHHTHH   Probability = 0.000727037502    [Unsuccessful termination]
    Event:  HHHTHT   Probability = 0.000718365087    [Unsuccessful termination]
    Event:  HHHTTH   Probability = 0.001771373159    [Unsuccessful termination]
    Event:  HHHTTT   Probability = 0.001750243459    [Successful termination]
    Event:  HHTHHH   Probability = 0.002843674515    [Unsuccessful termination]
    Event:  HHTHHT   Probability = 0.002809753944    [Unsuccessful termination]
    Event:  HHTHTH   Probability = 0.006928402856    [Unsuccessful termination]
    Event:  HHTHTT   Probability = 0.006845757892    [Successful termination]
    Event:  HHTT     Probability = 0.005901731793    [Successful termination]
    Event:  HT       Probability = 0.172353000000    [Successful termination]
    Event:  TH       Probability = 0.166353000000    [Successful termination]
    Event:  TTHH     Probability = 0.215445451793    [Successful termination]
    Event:  TTHTHH   Probability = 0.009579853361    [Successful termination]
    Event:  TTHTHT   Probability = 0.009465580756    [Unsuccessful termination]
    Event:  TTHTTH   Probability = 0.023340604924    [Unsuccessful termination]
    Event:  TTHTTT   Probability = 0.023062188166    [Unsuccessful termination]
    Event:  TTTHHH   Probability = 0.037469848214    [Successful termination]
    Event:  TTTHHT   Probability = 0.037022891774    [Unsuccessful termination]
    Event:  TTTHTH   Probability = 0.091292516782    [Unsuccessful termination]
    Event:  TTTHTT   Probability = 0.090203540438    [Unsuccessful termination]
    Event:  TTTTHH   Probability = 0.011382626641    [Unsuccessful termination]
    Event:  TTTTHT   Probability = 0.011246849783    [Unsuccessful termination]
    Event:  TTTTTH   Probability = 0.027732928827    [Unsuccessful termination]
    Event:  TTTTTT   Probability = 0.027402118543    [Unsuccessful termination]

+++ Extreme termination events                                                      (2)
    Most likely successful termination event     : TTHH
    Most unlikely successful termination event   : HHHTTT
    Most likely unsuccessful termination event   : TTTHTH
    Most unlikely unsuccessful termination event : HHHTHT

+++ Total probabilities                                                             (2)
    Probability of successful termination   : 0.615698886511
    Probability of unsuccessful termination : 0.384301113489

+++ Computation without the tree                                                    (2)
    Probability of successful termination   :  0.615698886511
    Probability of unsuccessful termination :  0.384301113489
```

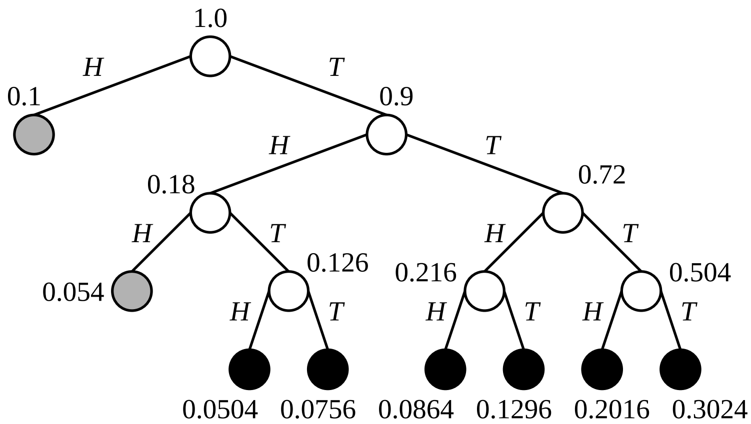
Submit one C/C++ file. Do not use STL data structures. Do not use global/static variables.
Write your name, roll number, and PC number as a comment near the beginning of your code.

You are given n biased coins with heads occurring with probabilities $p_0, p_1, p_2, \ldots, p_{n-1}$. You keep on tossing the coins in the given order (once each), and stop as soon as one of the following two events happens.

1. You get more heads than tails.
2. You have tossed all of the n coins.

Both the events may happen together (for odd n). You say that your experiment succeeds if Event 1 happens (irrespective of whether Event 2 happens together or not).

Model this random process as a binary tree. Each left-child link stands for the occurrence of a head H, and each right-child link stands for the occurrence of a tail T. Each node in the tree stores the probability of the toss outcomes so far. The following figure shows the probability tree for four coins with head probabilities $0.1, 0.2, 0.3, 0.4$. The leaf nodes represent the termination of the random process. The gray leaves stand for successful termination, and the black leaves stand for unsuccessful termination. In this example, the success probability is $0.1 + 0.054 = 0.154$, and the failure probability is $0.0504 + 0.0756 + 0.0864 + 0.1296 + 0.2016 + 0.3024 = 0.846$. The sum of these two probabilities must be 1.



Part 1: Tree data type **(4)**

Define a data type to store a node in the probability tree (a binary tree). Each node should consist of a floating-point (data type `double`) probability, two child pointers (left and right), optionally a parent pointer (may be helpful in Part 3), and nothing else. Only probabilities are stored in the nodes. Do *not* store the events (like TTH) anywhere in the tree. The node for any event can be reached by following appropriate pointers. Do *not* store, in any leaf node, whether this is a case of successful termination or not. In the remaining parts, you work only with the tree or the input array P.

Part 2: Build the tree **(12)**

Write a function *buildtree* that takes the array P of n head probabilities as input. The function recursively creates the entire probability tree, and returns a pointer to the root node of the tree.

Part 3: Tree functions **(4×3)**

Write functions that take as input the tree built in Part 2, and does the following (also see sample output).

allevents Print all termination events along with their respective probabilities and the information whether these are cases of successful or unsuccessful termination.

extremeevents Find the terminating events with minimum and maximum probabilities in both the cases of successful and unsuccessful termination.

successprob Compute the total probabilities of successful and unsuccessful termination.

Part 4: Success probability without the tree **(8)**

Write a function *notreecomp* that takes as input only the array P of n head probabilities (and not the tree built in Part 2), and computes the probabilities of successful and unsuccessful termination. The tree of Part 2 contains $\Theta(2^n)$ nodes, so the functions of Part 3 take time exponential in n. The complexity of *notreecomp* should be polynomial in n. More specifically, it should run in $O(n^2)$ time, and use only $O(n)$ extra space.

Part 5: The main function **(4)**

The user enters n followed by the probabilities $p_0, p_1, p_2, \ldots, p_{n-1}$. Store these probabilities in an array P of **`double`** variables. Call *buildtree* (Part 2) to build the probability tree. Call the functions of Part 3 one by one. Finally, call *notreecomp* of Part 4, and print the success and failure probabilities.

Output **(10)**

```
n = 6
Head probabilities: 0.139 0.799 0.271 0.620 0.778 0.119

+++ All termination events                                                        (4)
    Event:  H         Probability = 0.139000000000    [Successful termination]
    Event:  THH       Probability = 0.186431469000    [Successful termination]
    Event:  THTHH     Probability = 0.241907172653    [Successful termination]
    Event:  THTHTH    Probability = 0.008214272091    [Unsuccessful termination]
    Event:  THTHTT    Probability = 0.060813224475    [Unsuccessful termination]
    Event:  THTTHH    Probability = 0.017643616689    [Unsuccessful termination]
    Event:  THTTHT    Probability = 0.130622069776    [Unsuccessful termination]
    Event:  THTTTH    Probability = 0.005034553863    [Unsuccessful termination]
    Event:  THTTTT    Probability = 0.037272621453    [Unsuccessful termination]
    Event:  TTHHH     Probability = 0.022622457773    [Successful termination]
    Event:  TTHHTH    Probability = 0.000768174922    [Unsuccessful termination]
    Event:  TTHHTT    Probability = 0.005687076525    [Unsuccessful termination]
    Event:  TTHTHH    Probability = 0.001649979904    [Unsuccessful termination]
    Event:  TTHTHT    Probability = 0.012215397441    [Unsuccessful termination]
    Event:  TTHTTH    Probability = 0.000470816888    [Unsuccessful termination]
    Event:  TTHTTT    Probability = 0.003485627547    [Unsuccessful termination]
    Event:  TTTHHH    Probability = 0.007241774296    [Unsuccessful termination]
    Event:  TTTHHT    Probability = 0.053613471891    [Unsuccessful termination]
    Event:  TTTHTH    Probability = 0.002066418887    [Unsuccessful termination]
    Event:  TTTHTT    Probability = 0.015298445707    [Unsuccessful termination]
    Event:  TTTTHH    Probability = 0.004438506827    [Unsuccessful termination]
    Event:  TTTTHT    Probability = 0.032859869868    [Unsuccessful termination]
    Event:  TTTTTH    Probability = 0.001266514801    [Unsuccessful termination]
    Event:  TTTTTT    Probability = 0.009376466723    [Unsuccessful termination]

+++ Extreme termination events                                                    (2)
    Most likely successful termination event      : THTHH
    Most unlikely successful termination event    : TTHHH
    Most likely unsuccessful termination event    : THTTHT
    Most unlikely unsuccessful termination event  : TTHTTH

+++ Total probabilities                                                           (2)
    Probability of successful termination   : 0.589961099426
    Probability of unsuccessful termination : 0.410038900574

+++ Computation without the tree                                                  (2)
    Probability of successful termination   : 0.589961099426
    Probability of unsuccessful termination : 0.410038900574
```

Submit one C/C++ file. Do not use STL data structures. Do not use global/static variables.
Write your name, roll number, and PC number as a comment near the beginning of your code.