STATISTICAL STATIC TIMING ANALYSIS

NORTH DAKOTA STATE UNIVERSITY

Overview

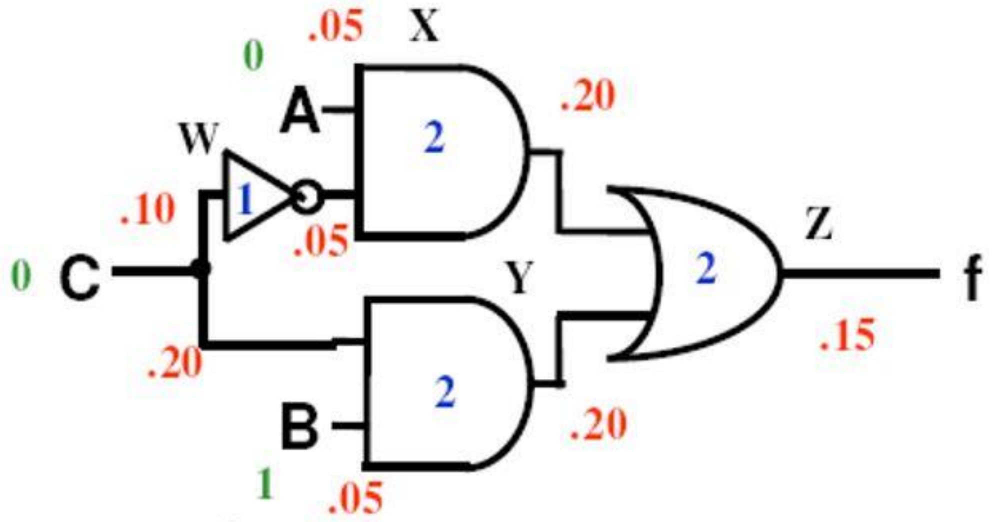
- One-Path Based Approach
- One-Block Based Approach
- Monte Carlo Simulation

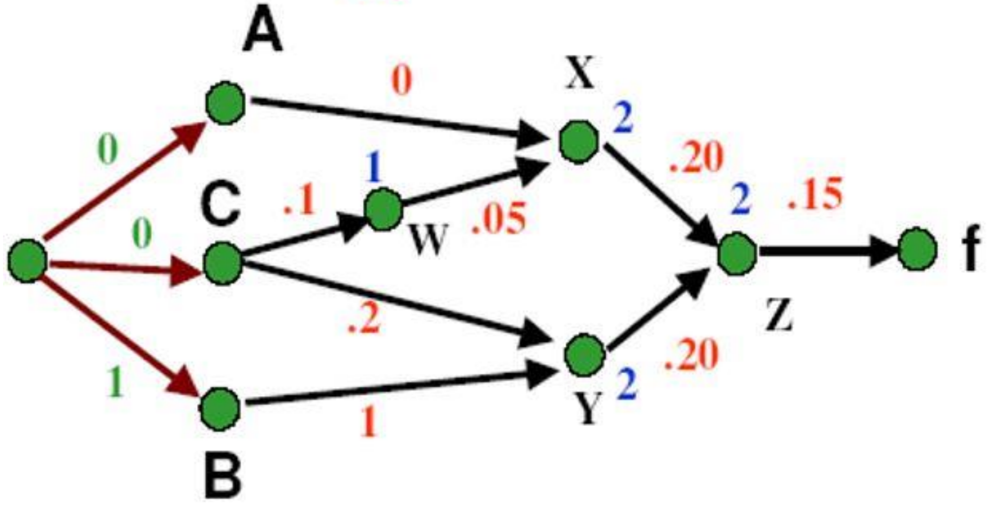
One-Path Based Approach

- Delay of each path is calculated (individually, source to sink)
- Delays calculated using gate delays and wire delays
- Path delay = sum(node delay)
- Critical path delay = max(path delay)
- Gate and path delay present variances
- Depth first search
- Problem: very expensive, can't EFFICIENTLY identify all likely critical paths (look at block based)

One-Path Based Approach

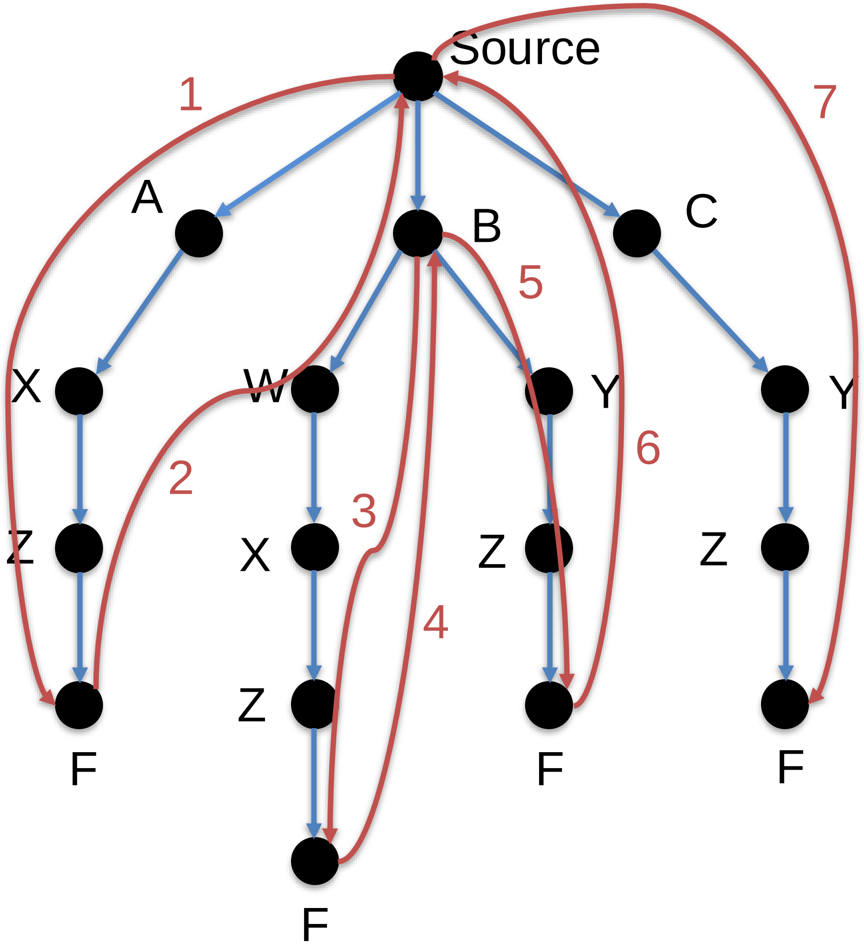
- Delay of the entire circuit can be analyzed by using basic calculations
- Graph is traversed
- Paths of interest should be defined before running (very expensive otherwise)
- Essentially is a “longest path algorithm”
- NP-Hard, NP-Complete





One-Path Based Approach

Depth First Search



One-Path Based Approach

Path-based STA

- Path enumeration
  - List all paths, output path delays
- Algorithms

```
search (path P, delay d) {
    n = last node in P;
    if ( there are no successor nodes to n )
        Output path P, delay d;    /* All paths end at sink */
    else {
        foreach (node s in succ(n) ) {
            search ( P+s, d+delay(n,s) );
        }
    }
}
```

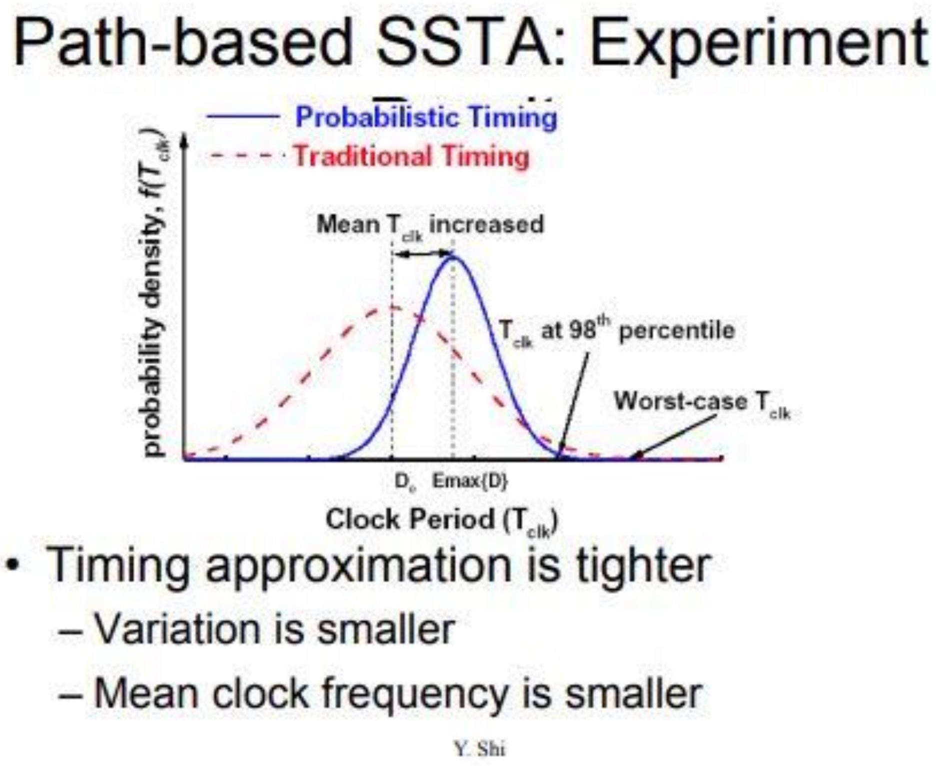
Add one more node to the end of the path and recurse

- It works, but what's the problem?

Y. Shi

One-Path Based Approach

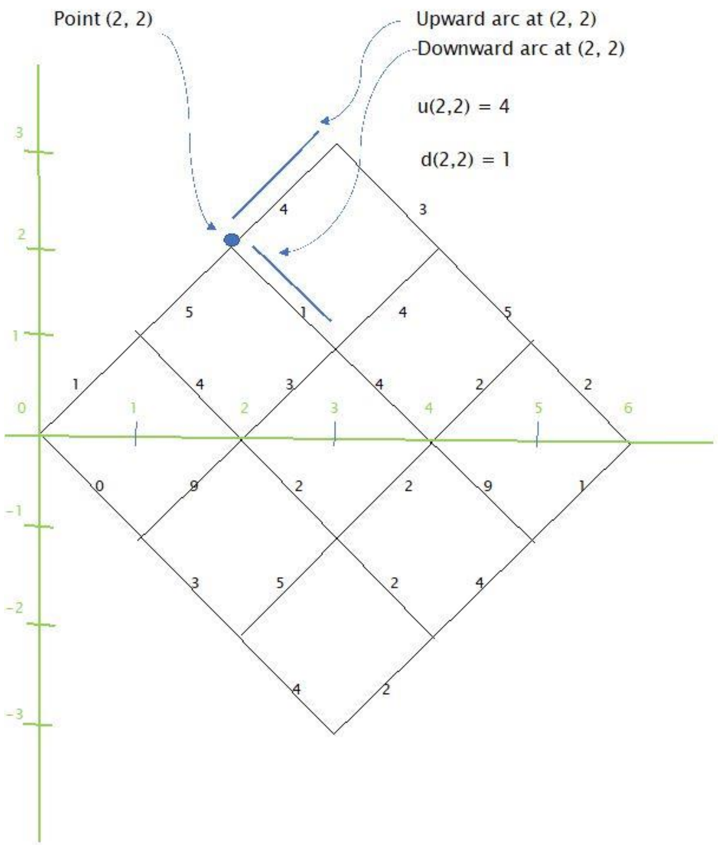
- Approximate the variance of the path delays
- Variance caused by gates and paths (degradation, electron migration, temperature, device fatigue, etc.)
- These can be represented as RVs (explained later)

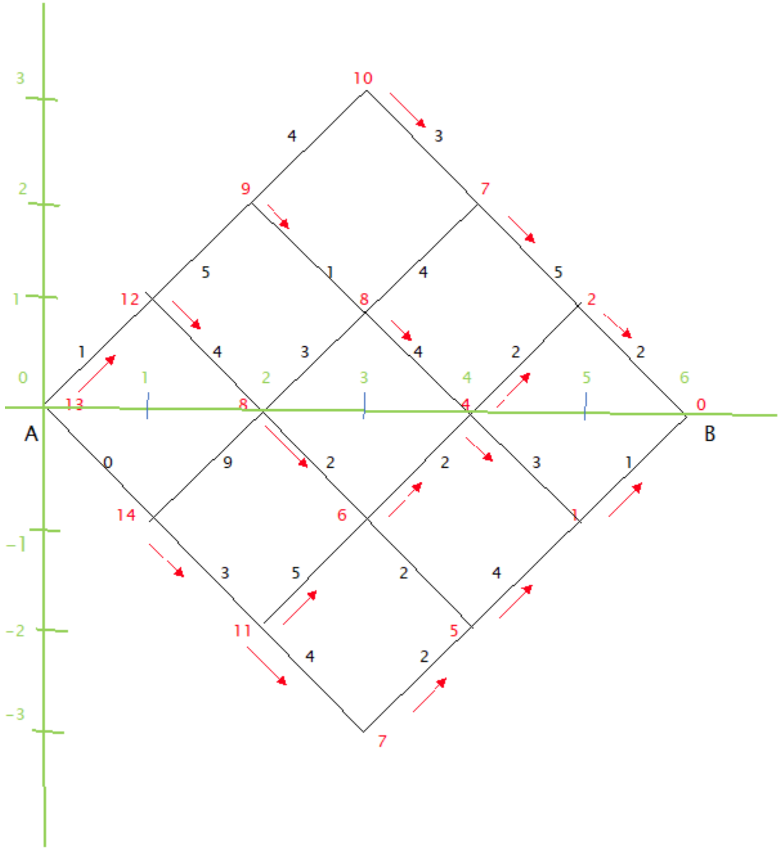


One-Block Based Approach

- Goal of this approach is to find AT (arrival time) and RAT (required arrival time)
- Used to evaluate all paths simultaneously
- Delay PDF of entire circuit calculated at the end of traversal
- AT (CDF) and Gate Delays (PDF) are modeled as RVs (similar to path and gate delays)

One-Block Based Approach





One-Block Based Approach

- What the previous slide shows us
  - We wish to find the shortest path (block based finds longest, but approach is similar)
  - We move from right (sink) to left (source)
  - Edges contain weights (delays)
  - Key idea of algorithm is similar to what we wish to achieve

One-Block Based Approach

- Algorithm performs these calculations recursively on all nodes
- Only difference is longest paths are “optimal”
- Maximum length paths (critical paths) are found

Backward dynamic programming formulation

u(x, y) : cost of the upward arc at (x, y)

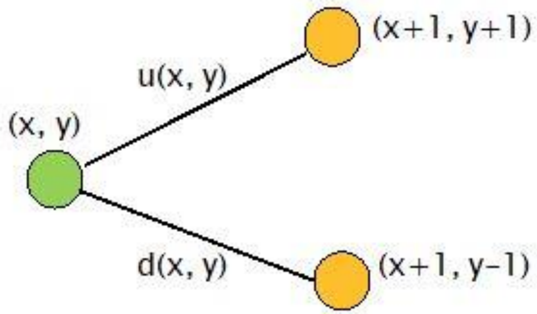
d(x, y) : cost of the downward arc at (x, y)

p(x, y) : policy at (x, y), go up or down or both

Optimality function

s(x, y): length of the shortest path from node (x, y) to node (6, 0).

Recurrence



$$s(x, y) = \min \begin{cases} u(x, y) + s(x+1, y+1) \\ d(x, y) + s(x+1, y-1) \end{cases}$$

One-Block Based Approach

- Similar to previous formulation
- In this case, we traverse forward, or “left to right”
- “Source to Sink”

Forward dynamic programming formulation

u(x, y) : cost of the upward arc at (x, y)

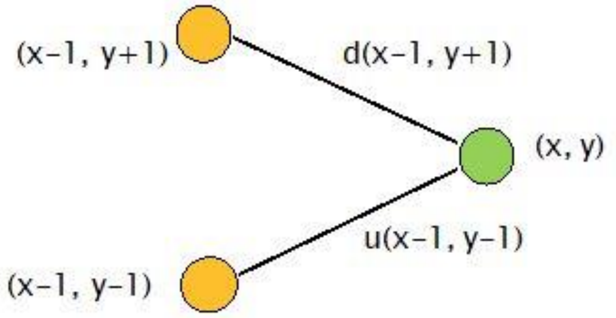
d(x, y) : cost of the downward arc at (x, y)

p(x, y) : policy at (x, y), arrive up or down or both

Optimality function

s(x, y): length of the shortest path from node (0, 0) to (x, y).

Recurrence



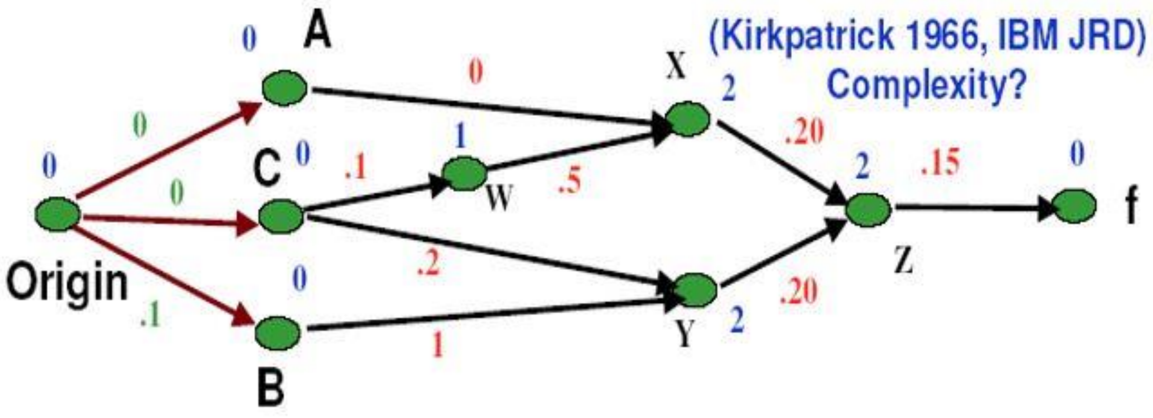
$$s(x, y) = \min \begin{cases} u(x-1, y-1) + s(x-1, y-1) \\ d(x-1, y+1) + s(x-1, y+1) \end{cases}$$

One-Block Based Approach

- How does this apply to One-Block Based?
  - Recursive calls are made from source node to sink node
  - Delay (or weight) at each node is calculated, with maximal delays/paths taking precedence
  - Critical path(s) will be identified
  - AT computed
  - Timing slack found by going backwards through DAG

One-Block Based Approach

AT Computation illustrated



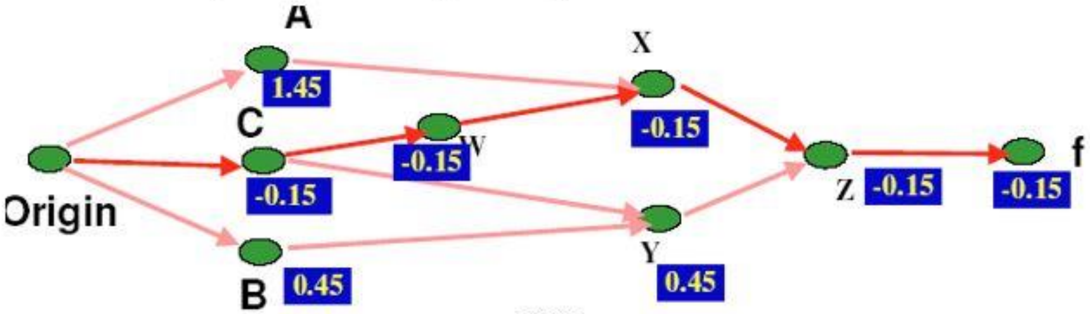

```
Forward-prop(W){
    For each vertex v  in W{
        For each edge <v,w> from v{
            Final-delay(w)=max(Final-delay(w), delay(v)+delay(w)+delay(<v,w>))
            If all incoming edges of w have been traversed
            Add w to W;
        }
    }
}
```

Y. Shi

One-Block Based Approach

Timing Slack

- Each node: AT & RAT
- Timing slack: RAT-AT
  - Reflects criticality of a node
  - Negative: timing violation, node is on critical path
  - Positive: extra timing budget
    - Optimize for power/area/robustness
    - Slack distribution is KEY for timing optimization!!
- An example: assuming RAT(f)=5.80



Y. Shi

Monte Carlo Simulation

- What is Monte Carlo Simulation?
  - Method of probability analysis to determine possible outcomes
  - Works by constructing a mathematical model of the considered situation
  - Simulation is run for any uncertain aspects of the mathematical model
  - Different RVs are put into uncertain parts until we have enough to plot on a probability distribution curve

Monte Carlo Simulation

- Powerful tool for numerical estimation
- Used for statistical estimations in SSTA
- Method is highly scalable and parallelizable
- Gives actual samples of different statistical outcomes, not just estimates of distributions or sensitivities
- Problem is it’s too slow and very expensive
- Typically used to validate accuracy of STA tools
- Isn’t used to implement practical SSTA tools

Monte Carlo Simulation

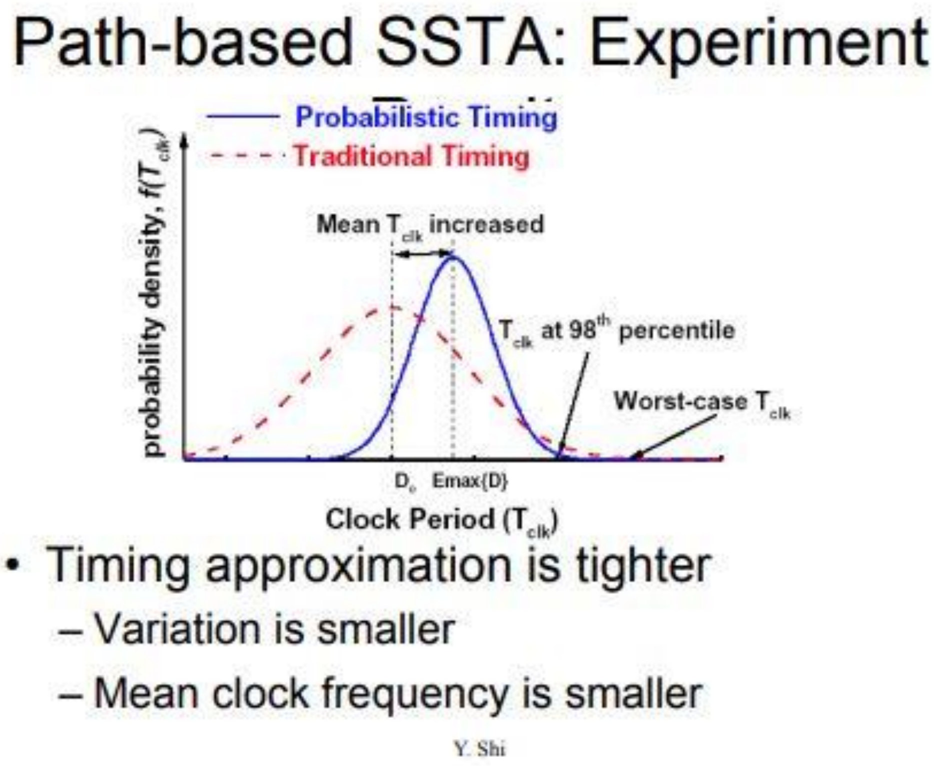
- Suppose $P = \{P^{(1)}, P^{(2)}, \ldots, P^{(d)}\}$
- Joint Probability Distribution Function
  - $\phi_d(P) = R \xrightarrow{d} R$
  - Each $P^{(\mathrm{i})}$ represents a process parameter (such as gate length, oxide thickness, interconnect dimension variations, etc.)
- $E[D^m(P)] = \int_{R^d} D^m(P)\phi(P)dP$ Where D(P) is the critical delay of a circuit as a function of the process parameters

Monte Carlo Simulation

- What is this simulation looking to perform with the equations?
  - Probability analysis to determine possible outcomes
  - Used to determine range of possibilities and their probability of occurrence
  - What ranges of time delays can we see, and what are the probabilities they occur
  - Things such as variance of path and gate delays produce these ranges of outcomes
  - Process parameters can affect things such as gate and path delays explained previously
- In a broad view, SSTA is used to calculate time delay distribution, but the algorithms covered previously perform this computation to define the distribution

Monte Carlo Simulation

- As was shown previously, we want to determine the probability density (some PDF) using path based SSTA
- Monte Carlo Simulation can be used to determine these probabilities
- Each run of the simulation can help plot a probability distribution curve



References

- https://users.ece.cmu.edu/~rutenbar/pdf/rutenbar-iccad08.pdf
- https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol43-4/paper18.pdf
- https://slideplayer.com/slide/16944683/
- https://www.slideserve.com/zenaida-english/statistical-static-timing-analysis
- CSCI 741 – Algorithm Analysis (Dynamic Programming Part I)