**Question1:**

**a). Formulate the task**

* Create the digraph called G:

1. G = (F {s, t} E)
2. (k = total number of forests; Y = total number of years; i = the index from 1 to k; j = the index from j to Y)
3. Representations for the type of vertices:
   1. s = Source
   2. F = The forests which is {forest 1 to forest k} since k is the total number of forests
   3. = The year, which the trees will be matured from F (all the forests). The year would be declared as {mature\_ year 1 to mature\_ year Y} since Y is the total number of years.
   4. = The year, which the matured trees can be cut down and sell. The year would be declared as {sell\_ year 1 to sell\_ year Y} since Y is the total number of years.
   5. t = Sink
4. Representations for the type of edges:

- (the edge between s and F): The total trees in each forest

Capacity in : ()

- (the edge between F and ): The total amount of matured trees of each forest in each year

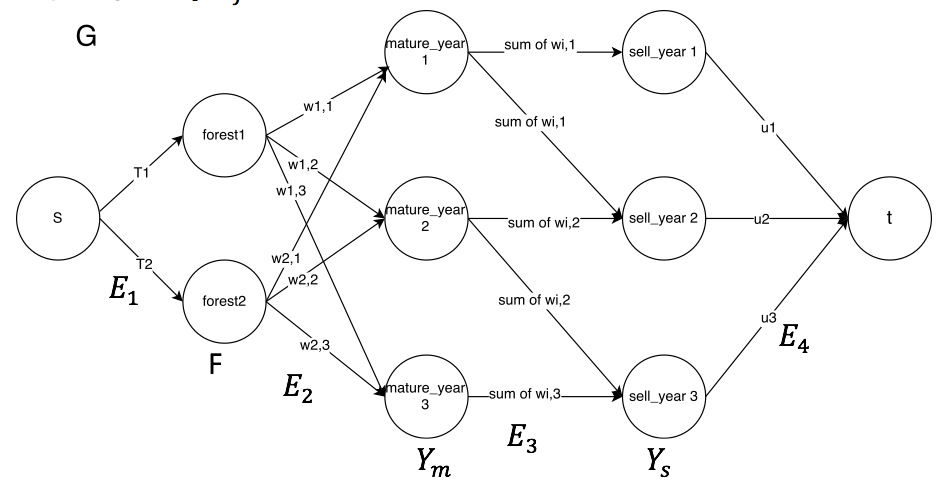
Capacity in :

- (the edge between and ): The amount of matured-trees from the matured year to the available selling years, we can also says each tree matures in year j can only be sold in that year (j) or the - 1 years afterwards.

Capacity in : The sum of the of all the forests in each mature-year (), which goes to the available sell \_year ().

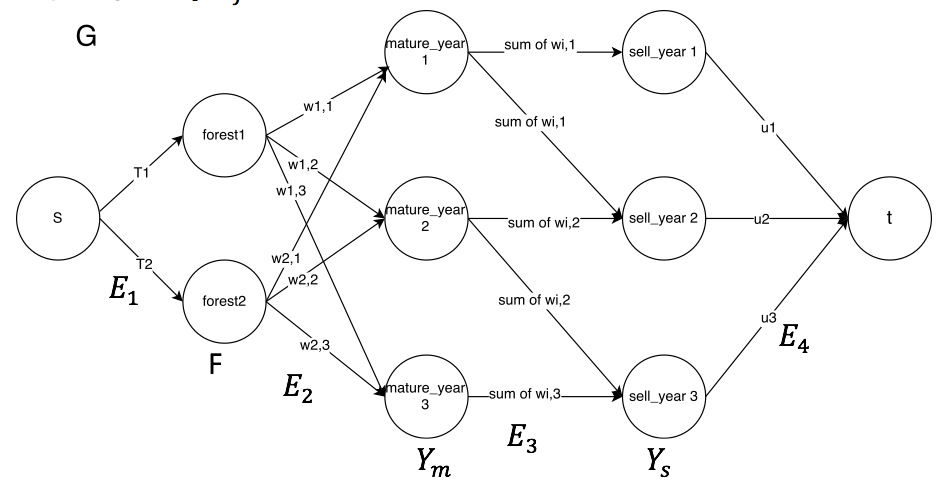
- (the edge between and t): The amount of the total sell-matured-trees from each year.

Capacity in :



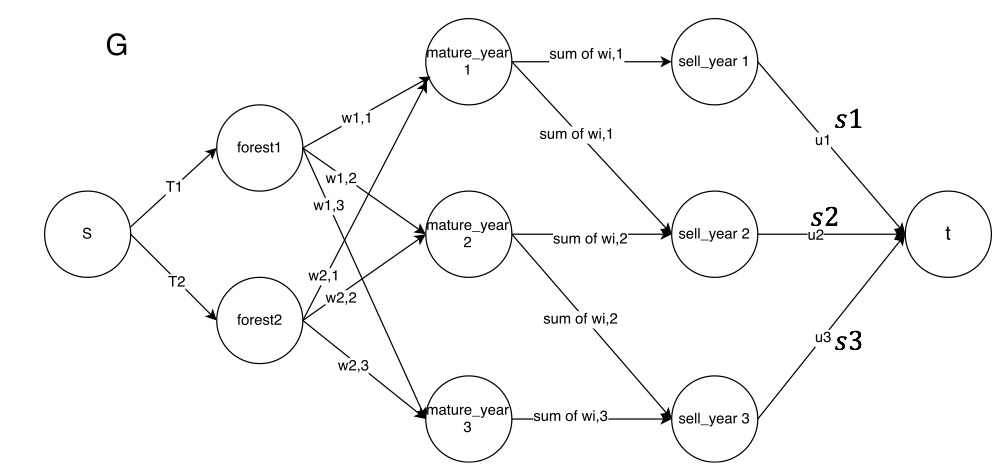
**b). Prove the algorithm is correct:**

* Integrality: If all capacities are integers then every flow value f(e) and every residual capacities remains an integer throughout the algorithm.



**Proof (part I and part II):**

* **Part I:** 
  1. Purpose: If the maximum number of Christmas trees sold is , we also have to prove that there exists a max flow .

 1). The flow of the edges from sell\_ year nodes to the sink (t) is represented as the schedule that sells Christmas trees. We mark the flow from sell\_ year 1 to t as s1, the flow from sell\_ year 2 to t as s2 and the sell\_ year 3 to t as s3. Since we have assumed the max schedule that sells the Christmas trees is . Hence, the sum of the flows (s1 + s2 + s3) is .

2). There are two edges going from the source to forest 1 and forest 2. The flow of the edges between source (s) to the forest nodes (F) is represented as the number of the trees (the mature\_ trees can be sold) in each forest. We mark the flow between s to forest 1 as fo1 and the flow between s to forest 2 as fo2.

../../../../../../Downloads/Untitled%20Diagram%20(1)

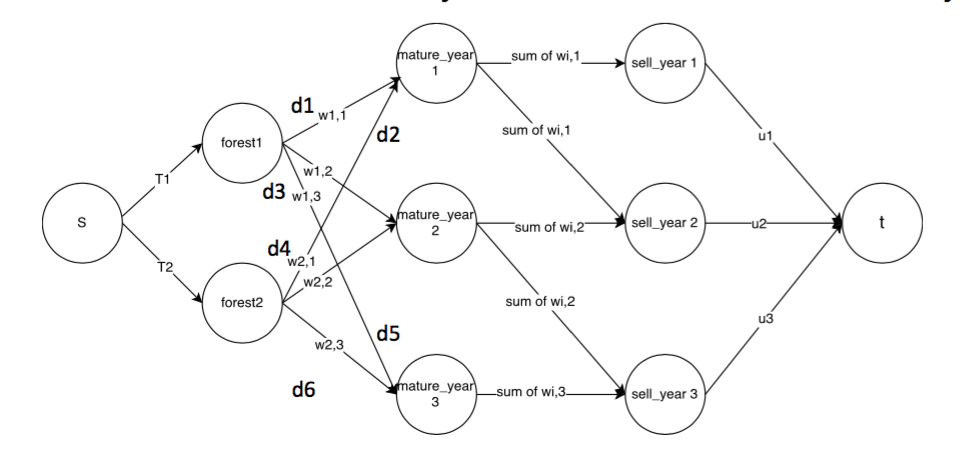
In the flows definition, the value of a flow is: .

Since we have assumed the max value of the schedule is , therefore, the sum of the flow between s to the forests (fo1 + fo2) suppose to be as well, which is the sum of the flows between sell\_ years to t (s1 + s2 + s3).

“ fo1 + fo2 = s1 + s2 + s3 = max schedule ”

Since the max schedule f is assumed, the capacity of the edge between s to forest nodes should be holds by the flow definition: . In our graph the definition should be .

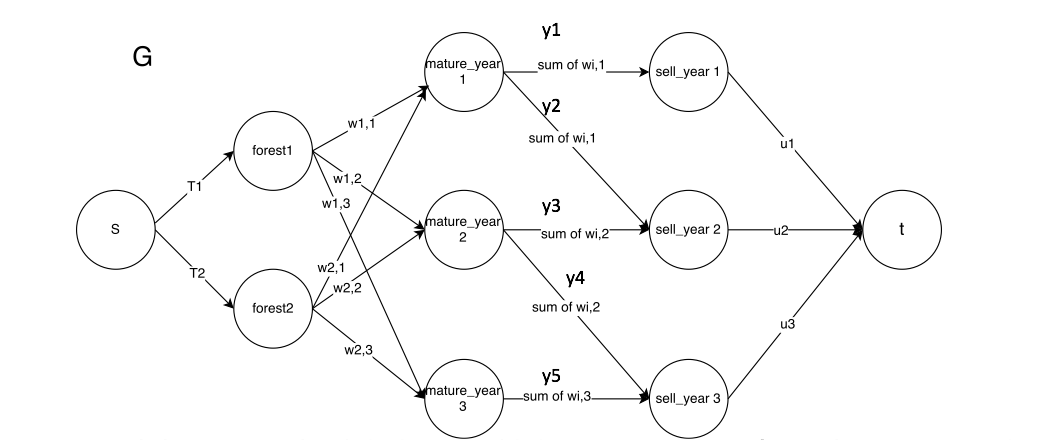
3). The flow of the edges between the forest nodes to the mature\_ year nodes () is represented as the number of forest i’s trees mature in year j. We mark the flow between forest 1 to mature\_ year 1 as d1, forest2 to mature\_ year 2 as d2, forest3 to mature\_ year 3 as d3, forest4 to mature\_ year 4 as d4, forest5 to mature\_ year 5 as d5 and forest6 to mature\_ year 6 as d6.



Since the max schedule is , therefore:

* 1. .
  2. The capacity of the edge between forests to mature\_ years holds.
  3. We have to assume the max schedule is correct.
  4. For each v : = in the flows conservation rule.
  5. The conservation between flow and flow holds, since:
  6. =
     1. and
  7. =

4). The flow of the edge from the mature\_ year nodes to the sell\_ year nodes is represented as the number of the matured trees be allowed to cut down and sell in the year j or years afterwards. We mark the flow from mature\_ year 1 to sell\_ year 1 as , mature\_ year1 to sell\_ year 2 as y2, mature\_ year 2 to sell\_ year 1 as y3, mature\_ year 2 to sell\_ year 3 as y4 and mature\_ year 3 to sell\_ year 3 as y5.



We have assumed the max schedule is and it is correct. Therefore, the conservation between flow d and flow y holds, since:

=

and

=

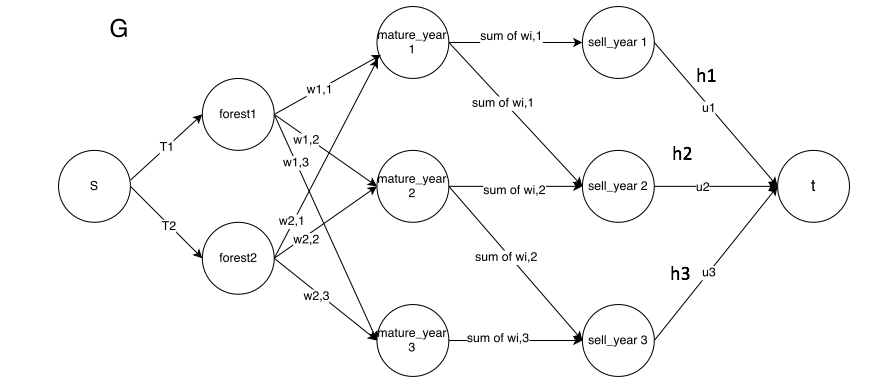
and

=

Additionally, the capacity of the edge between mature\_ year nodes to the sell\_ year nodes would hold since:

5). The flow of the edges between sell\_ year nodes to the sink (t) is represented as the number of the trees that be harvested in year j. we mark the flow of the edge from

sell\_ year 1 to t as , sell\_ year 2 to t as and sell\_ year 3 to t as .



Since the assumption that max schedule is . Hence the capacity of the edge between sell\_ year nodes to t has to hold when:

.

In the flows conservation rule, the conservation between flow and flow h should hold as well since:

=

and

=

and

=

* **Part II:**

(each flow’s meaning and the signs between the different type of set-nodes has been defined in part I and those flows would still be used for part II.)

Purpose: if assume is the value of the max flow then we have to prove there exist a schedule that sells Christmas trees as well.

Hence, the capacities () of the edge (E1, E2, E3, E4) should hold ().

The conservation of the s-t flow for each set of nodes (F, ) should hold as well

( v : = )

Since there exists a max flow with value , therefore, the sum of the flows of E4, which is has to be as well. Additionally, the sum of the flows of E1, which is also has to be since the definition of the flows:

* **Conclusion for the proof:**

In part I we have proved that there exists a max flow with value via holding the capacity for the edges and the conservation for the nodes since we assume the max schedule that sells number of Christmas trees. In part II, we prove there exists a max schedule that sells trees through assuming the value of max flow exists. Hence, our algorithm is correct since we have proved the algorithm’s correctness in both directions.

**Question2:**

**a). Formulate the task**

* Create the digraph called G:

1. G = (F {s, t} E)
2. (k = total number of forests; Y = total number of years; i = the index from 1 to k; j = the index from j to Y)
3. Representations for the type of vertices:
   1. s = Source
   2. F = The forests which is {forest 1 to forest k} since k is the total number of forests
   3. = The year, which the trees will be matured from F (all the forests). The year would be declared as {mature\_ year 1 to mature\_ year Y} since Y is the total number of years.
   4. = The year, which the matured trees can be cut down and sell. The year would be declared as {sell\_ year 1 to sell\_ year Y} since Y is the total number of years.
   5. t = Sink
4. Representations for the type of edges:

(we make up the fake value of the capacities)

- (the edge between s and F): The total trees in each forest

Capacity in :

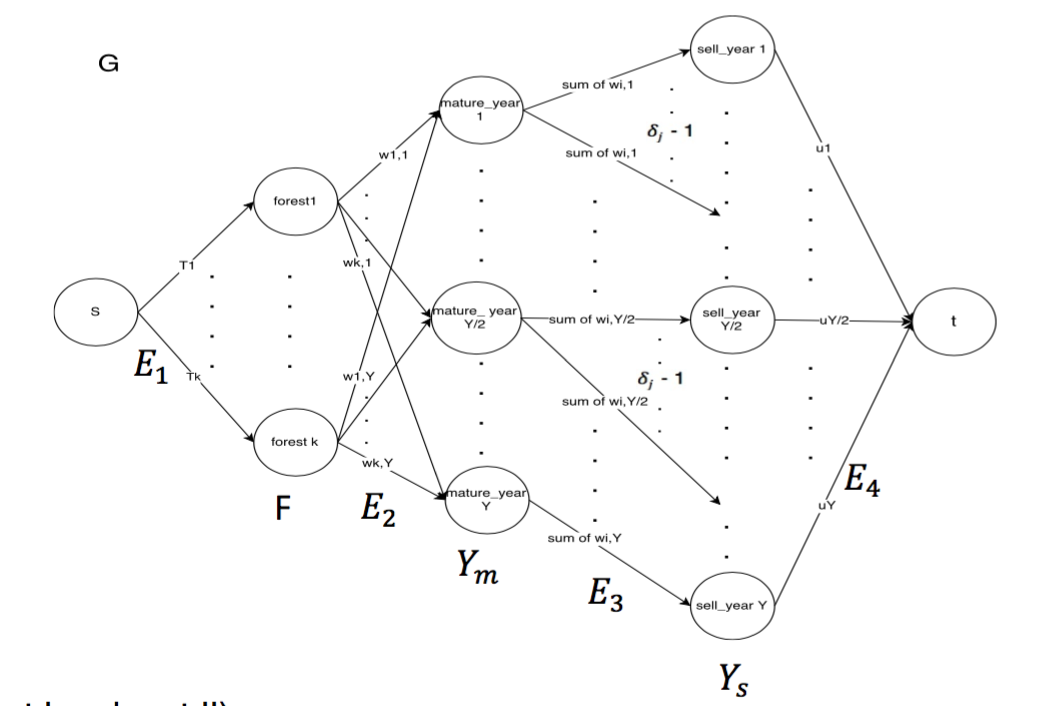
- (the edge between F and ): The total amount of matured trees of each forest in each year

Capacity in :

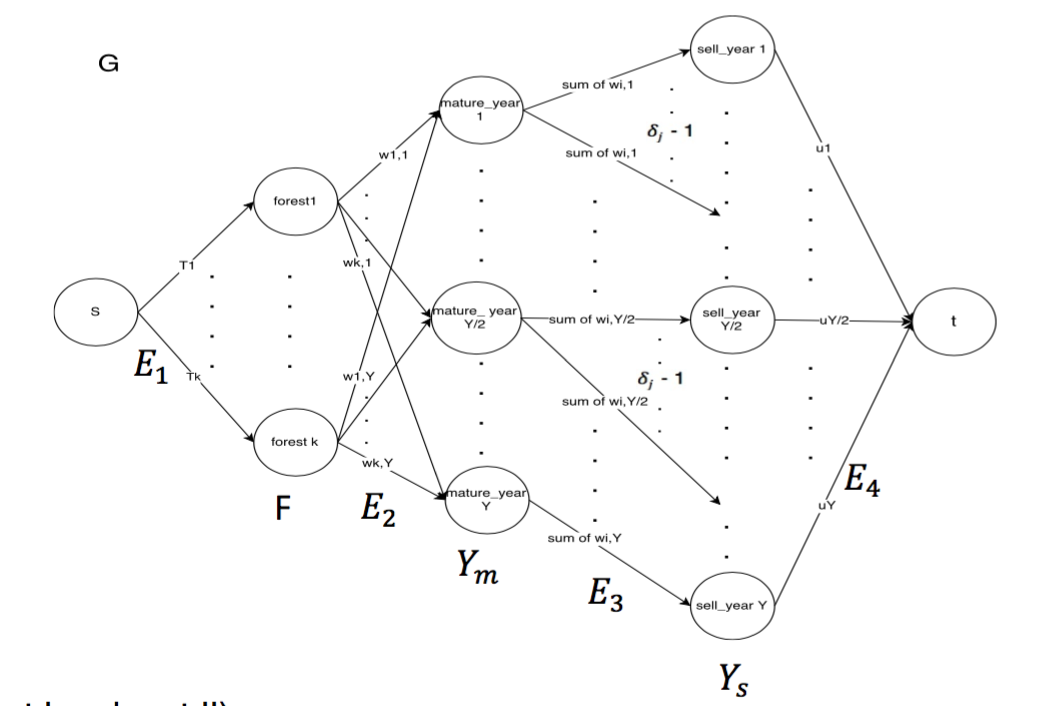
- (the edge between and ): The amount of matured-trees from the matured year to the available selling years, we can also says each tree matures in year j can only be sold in that year (j) or the - 1 years afterwards.

Capacity in : The sum of the of all the forests in each mature-year (), which goes to the available sell \_year ().

- (the edge between and t): The amount of the total sell-matured-trees from each year

 Capacity in :

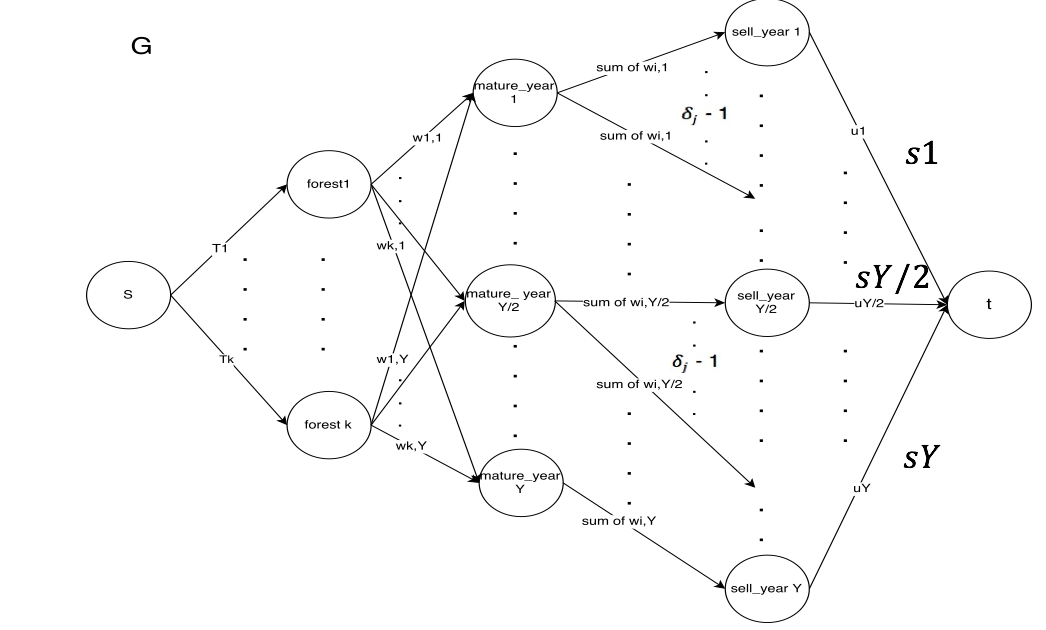
**b). Prove the algorithm is correct:**

* Integrality: If all capacities are integers then every flow value f(e) and every residual capacities remains an integer throughout the algorithm.

**Proof (part I and part II):**

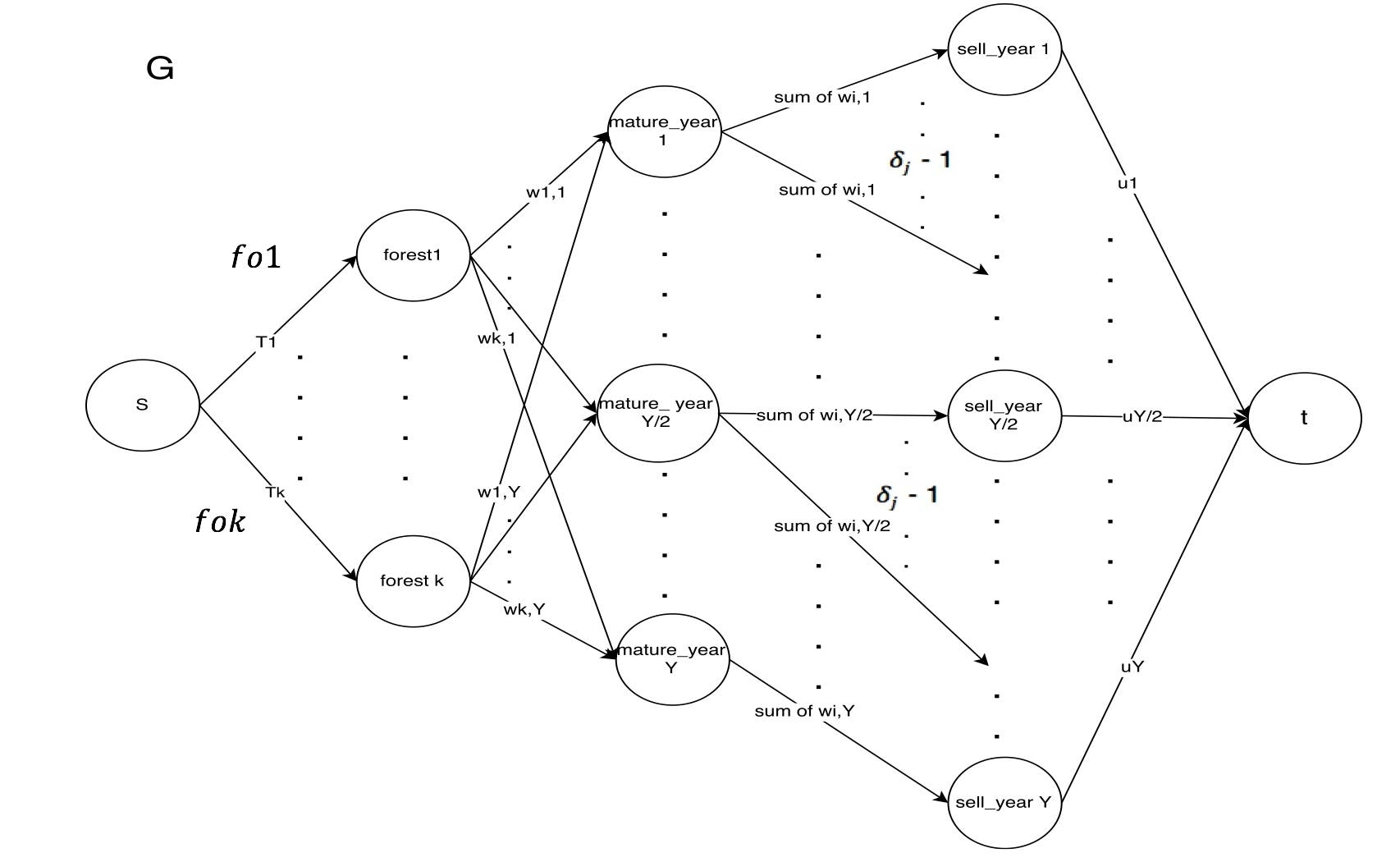
* **Part I:** 
  1. Purpose: If the maximum number of Christmas trees sold is , we also have to prove that there exists a max flow .

1). The flow of the edges from sell\_ year nodes to the sink (t) is represented as the schedule that sells Christmas trees. We mark the flow from sell\_ year 1 to t is s1 up to the flow from sell\_ year Y to t is sY. Since we have assumed the max schedule that sells the Christmas trees is . Hence, the sum of the flows (s1 + … + sY) is .



2). There are two edges go from the source to forest 1 to forest k. The flow of the edges between source (s) to the forest nodes (F) is represented as the number of the trees (the mature\_ trees can be sold) in each forest.

We mark the flow between s to forest 1 as fo1 up to the flow between s to forest k as fok.



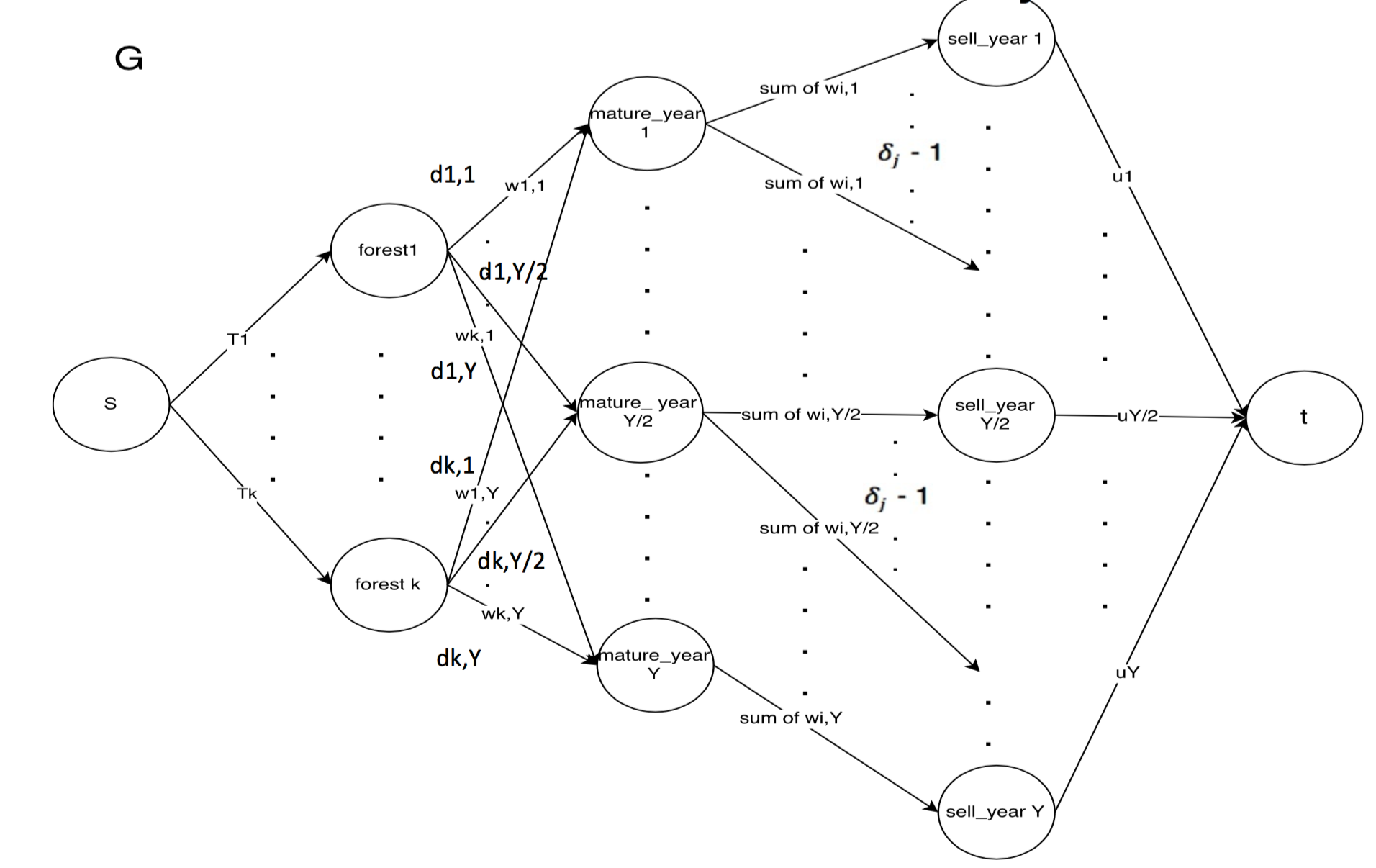
In the flows definition, the value of a flow is: .

Since we have assumed the max value of the schedule is , therefore, the sum of the flow between s to the forests (fo1 + … + fok) suppose to be as well, which is the sum of the flows between sell\_ years to t (s1 + … + sY).

“ fo1 + … + fok = s1 + … + sY = max schedule f ”

Since the max schedule is assumed, the capacity of the edge between s to forest nodes should be holds by the flow definition: . In our graph the definition should be .

3). The flow of the edges between the forest nodes to the mature\_ year nodes () is represented as the number of forest i’s trees mature in year j. We mark the flow between forest 1 to all of the mature\_ years from year1 to year Y as from d1,1 to d1,Y up to the flow between forest k to all of the mature\_ years as from dk,1 to dk,Y.

* 1. 

We have to assume the max schedule is correct.

Since the max schedule is , therefore:

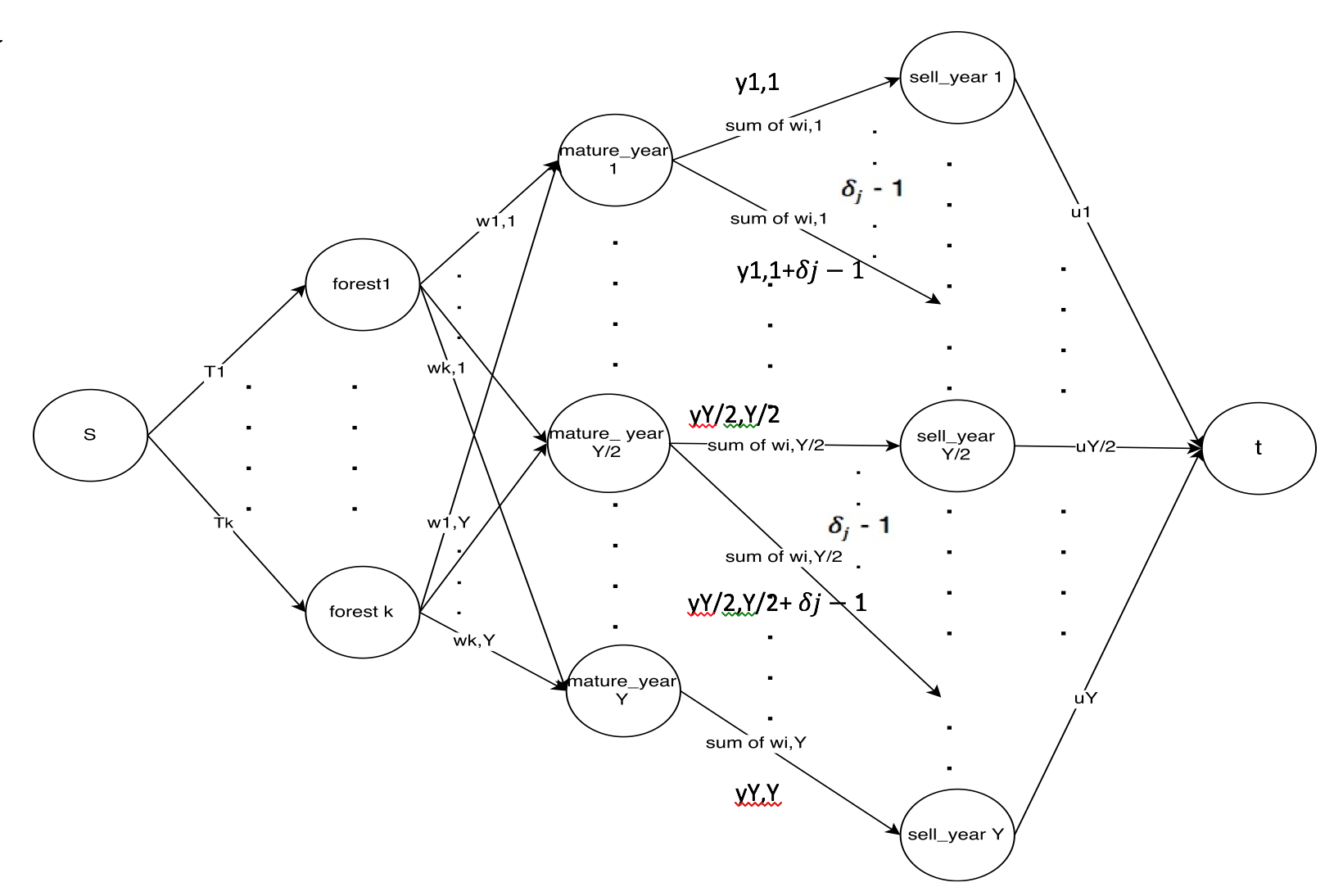
.

The capacity of the edge between forests to mature\_ years holds.

For each v : = in the flows conservation rule.

The conservation between flow and flow holds, since:

* 1. =

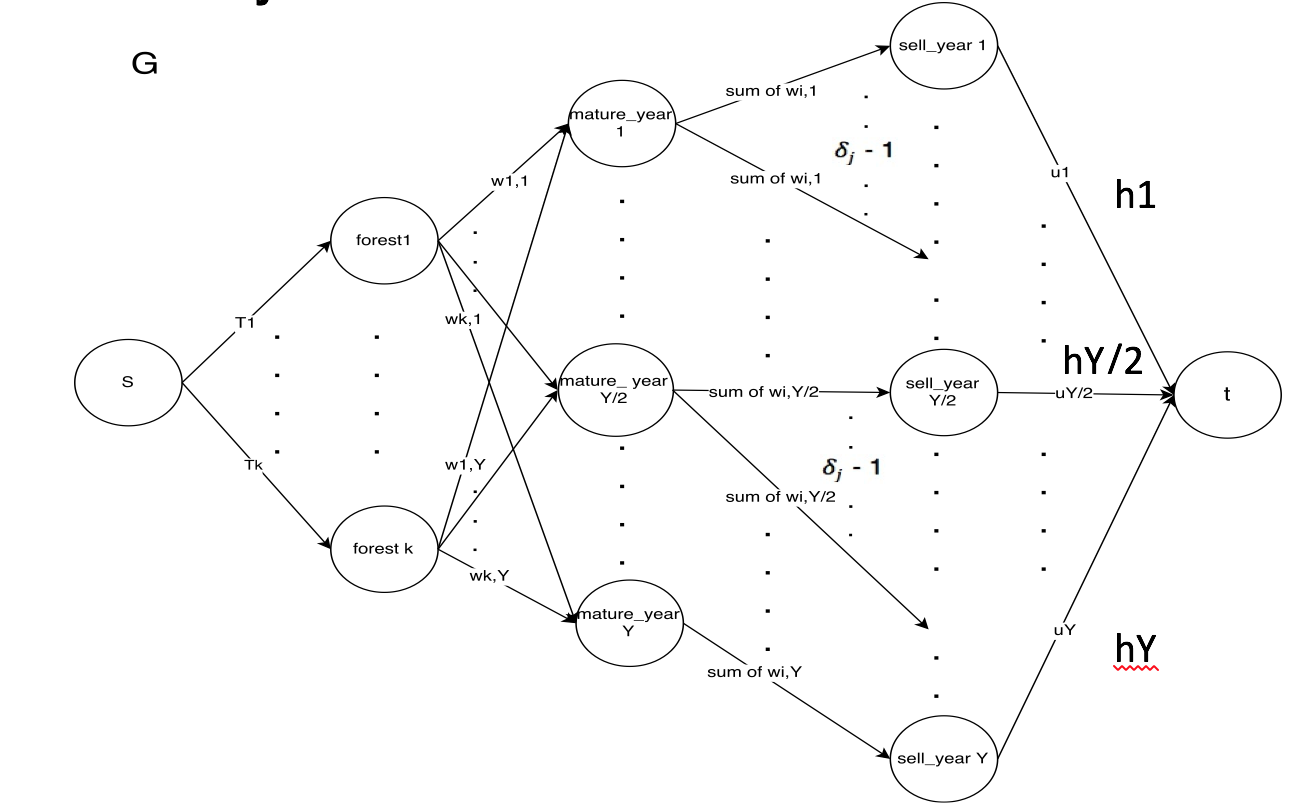
4). The flow of the edge from the mature\_ year nodes to the sell\_ year nodes is represented as the number of the matured trees be allowed to cut down and sell in the year j or years afterwards. We mark the flow from mature\_year1 to all the available sell\_ years as {y1,1 to y1,} up to the flow from mature\_ year Y to all the available sell\_ years as {yY,Y to yY,Y + }.

We have assumed the max schedule is and it is correct. Therefore, the conservation between flow d and flow y holds, since:

=

Additionally, the capacity of the edge between mature\_ year nodes to the sell\_ year nodes would hold since:

5). The flow of the edges between sell\_ year nodes to the sink (t) is represented as the number of the trees that be harvested in year j. we mark the flow of the edge from

sell\_ year 1 to t as up to sell\_ year Y to t as .

Since the assumption that max schedule is . Hence the capacity of the edge between sell\_ year nodes to t has to hold when:

.

In the flows conservation rule, the conservation between flow and flow h should hold as well since:

=

* **Part II:**

(each flow’s meaning and the signs between the different type of set-nodes has been defined in part I and those flows would still be used for part II.)

Purpose: if assume is the value of the max flow then we have to prove there exist a schedule that sells Christmas trees as well.

Hence, the capacities () of the edge (E1, E2, E3, E4) should hold ().

The conservation of the s-t flow for each set of nodes (F, ) should hold as well

( v : = )

Since there exists a max flow with value , therefore, the sum of the flows of E4, which is has to be as well. Additionally, the sum of the flows of E1, which is also has to be since the definition of the flows:

* **Conclusion for the proof:**

In part I we have proved that there exists a max flow with value via holding the capacity for the edges and the conservation for the nodes since we assume the max schedule that sells number of Christmas trees. In part II, we prove there exists a max schedule that sells trees through assuming the value of max flow exists. Hence, our algorithm is correct since we have proved the algorithm’s correctness in both directions.

**c). Upper bound of the algorithm:**

(The variables that have been defined in part a will be use for defining the upper bound of the time complexity for our algorithm).

We assume the max value of the flow is , and m is the number of all the edges in the graph .

* **The time complexity of Ford-Fulkerson O():**
* **The time complexity of setting up the graph:**

1). We firstly start from setting up the edges connection between the source (s) and the set of forest nodes (F). Hence, we just need to run with one for- loop with the range of k forests. The upper bound time complexity of the connection between s and F would be O(k).

2). We set up the edges connection between the set of forests (F) and all of the mature\_ years (). Since we have to set up each capacity () of the edge from each forest (1 to k) to all of the years (1 to Y). Hence, we have to do double for- loop with the outer range of k forests and the inner range of Y years. The upper bound time complexity of this operation would be O(kY).

3). We have to restore the capacity between the mature\_ year () and sell\_ year () since the capacity of the edge is the sum of the of all the forests in each mature-year () ), which goes to the available sell \_year (). Therefore, we should use double for-loop with the outer range of Y years and the inner range of k forests. The upper bound time complexity for this set-up is O(kY).

4). We set up the edge connection between the mature\_ years () to the sell\_ years(). Since the mature trees in each year sell to the available sell\_ years depends on (each year j connects to sell\_ year j to sell\_ year ), hence, we are using double for-loop with the outer range of Y years and the inner range of years. The time complexity for this operation should be O() since each trees in the mature\_ year can be cut down and sold out in year j and years afterwards depends on the value of the

5). The last set up connection is between the sell\_ years () to the sink (t). This is done by simply using a for-loop with the range of Y years to collectively go to the sink. Hence, the upper bound time complexity of this part would be O(Y).

* **Overall:**

We know {E1 + E2 + E3 + E4} set of edges include all the number of the edges in the graph, and we have declared the edge connection of each part in the graph. Hence, we convert from {E1 + E2 + E3 + E4} to {k + kY + + Y}. The Ford-Fulkerson’s time complexity would be converted from O() to O() in our graph. ( is the value of the maximum flow). The maximum value of can be infinite since the value can beyond Y years. Hence, we suppose to limit the ’s maximum value as Y years to prevent the algorithm might cause the extreme huge time complexity.

Additionally, the max value of the flow would be the max schedule that trees sold, which is . Since we know that the schedule that the trees sold would be the sum of the flows between the sell\_ years and the sink from sell\_ year 1 to sell\_ year Y. Hence, the value max schedule would also be . The would also represent as .

The task is asking for the maximum flow through our set-up graph. Hence, we implement Ford- Fulkerson in our graph to find the value of the max flow. Additionally, we need to consider that the ‘s constraint, so the upper bound of the edge connection between mature\_ year and sell\_ year would be from to O() (includes the part of Ford-Fulkerson’s time complexity as well). As a result, the total time complexity of the whole operations is:

* 1. O(k + ckY + + Y) + O()
  2. Which equals to the upper bound of:
     1. O( (kY + ) + )