MaxTemp algoritam je prilagođeni *Traversal* algoritam DF* (Santoro); u *Return* poruci čvorovi vraćaju svoju maksimalnu temperaturu, a pri primitku *Return* poruke čvor uspoređuje upravo primljenu temperaturu sa svojom maksimalnom temperaturom i zadržava onu višu.

Za "očitanje" početne temperature koristi se senzor *TemperatureSensor* implementiran unutar *sensor.py* skripte koji vraća nasumičnu vrijednost 0-100.

Pseudokod (posebno su označene linije dodane u već postojeći DF* kod):

```
- Status: S = {INITIATOR,IDLE,AVAILABLE,VISITED,DONE};
  S_INIT = {INITIATOR,IDLE}; S_TERM = {DONE}.
- Restrictions: R ;UI.

INITIATOR
      Spontaneously
      begin
            entry:= None;
            Unvisited:= N (x);
            next ⇐ Unvisited;
            send(T) to next;
            send(Visited) to N (x)-{next};
            become VISITED
      end

IDLE
      Receiving(T)
      begin
            Unvisited:= N (x);
            FIRST-VISIT;
      end

      Receiving(Visited)
      begin
            Unvisited:= N (x) – {sender};
            become AVAILABLE
      end

AVAILABLE
      Receiving(T)
      FIRST-VISIT;

      Receiving(Visited)
      begin
            Unvisited:= Unvisited – {sender};
      end


VISITED
      Receiving(Visited)
      begin
            Unvisited:= Unvisited –{sender};
            if next = sender then VISIT; endif
      end

      Receiving(T)
      begin
            Unvisited:= Unvisited –{sender};
```

```
            if next = sender then VISIT; endif
      end

      Receiving(Return)
      begin
            maxTemp:= max(MaxTemp, receivedMaxTemp)
            VISIT;
      end

Procedure FIRST-VISIT
begin
      entry:=sender;
      Unvisited:= Unvisited-{sender};

      if Unvisited = ∅ then
            next ⇐ Unvisited;
            send(T) to next;
            send(Visited) to N(x)−{entry,next};
            become VISITED;
      else
            send(Return, data=MaxTemp) to {entry};
            send(Visited) to N(x)−{entry};
            become DONE;
      endif
end

Procedure VISIT
begin
      if Unvisited = ∅ then
            next ⇐ Unvisited;
            send(T) to next;
      else
            if entry!=None then send(Return, data=MaxTemp) to entry; endif
            become DONE;
      endif
end
```

Analiza algoritma

Vremenska složenost algoritma i broj poruka jednaki su kao u DF* algoritmu.

n - broj čvorova

m - broj bridova

f* - broj čvorova koji pri prvom primitku *T* poruke nemaju neposjećenih susjeda

Vremenska složenost: **T [**MaxTemp**]** = 2n − 2
Broj poruka: **M [**MaxTemp**]** = 4m − 2n + f* + 1