

- EKINTZA HAUTAKETAREN PROBLEMA

Prozedura-hautesle onak (froga badute):

- Lehenen amaitzen duten ekintzak
- Beranduen hasten diren ekintzak

Osogarria

Hautatutako ekintza jadanik soluzio partzialean daudenekin teilakatzen ez bada

SoluzioaDa?

Soluzio partziala beti izango da soluzioa: kardinalitate maximoa + hautatutako ekintzen artean teilakapenik ez dago

Kodea

```
public class EkintzaPlanifikatzailea {
    public LinkedList<Ekintza> ekintzakHautatu(LinkedList<Ekintza>
ekintzak) {
        LinkedList<Ekintza> emaitza = new LinkedList<Ekintza>();
        LinkedList<Ekintza> ekintzenKopia = (LinkedList<Ekintza>)
ekintzak.clone();
        /* Ordenatu: Lehenen amaitzen dutenak --> Mergesort
erabiltzen da */
        Collections.sort(ekintzenKopia, new Comparator<Ekintza>() {
            public int compare(Ekintza lhs, Ekintza rhs) {
                if (lhs.getAmaieraOrdua().getTime() <
rhs.getAmaieraOrdua().getTime())
                    return -1;
                else if (lhs.getAmaieraOrdua().getTime() ==
rhs.getAmaieraOrdua().getTime())
                    return 0;
                else
                    return 1;
            }
        });
        /* Hautaketarekin hasi */
        Ekintza lehena = ekintzenKopia.removeFirst();
        emaitza.add(lehena);
        while(!ekintzenKopia.isEmpty()){
            Ekintza hautagaia = ekintzenKopia.removeFirst();

            if(hautagaia.getHasieraOrdua().getTime() >= lehena.getAmaieraOrdua
().getTime()){
                lehena = hautagaia;
                emaitza.add(lehena);
            }
        }
        return emaitza;
    }
}
```

```
public class Frogak {

    public static void main(String[] args) {
        LinkedList<Ekintza> ekintzak = new LinkedList<Ekintza>();
        Ekintza e1 = new Ekintza(StringToDate("2018-04-18
08:30:00"),StringToDate("2018-04-18 09:30:00"));
        Ekintza e2 = new Ekintza(StringToDate("2018-04-18
12:00:00"),StringToDate("2018-04-18 12:45:00"));
    }
}
```

```

        Ekintza e3 = new Ekintza(StringToDate("2018-04-18
11:30:00"),StringToDate("2018-04-18 12:15:00"));
        Ekintza e4 = new Ekintza(StringToDate("2018-04-18
11:00:00"),StringToDate("2018-04-18 12:30:00"));
        Ekintza e5 = new Ekintza(StringToDate("2018-04-18
10:00:00"),StringToDate("2018-04-18 10:50:00"));
        ekintzak.add(e1);ekintzak.add(e2);ekintzak.add(e3);
        ekintzak.add(e4);ekintzak.add(e5);
        EkintzaPlanifikatzailea ep = new EkintzaPlanifikatzailea();
        LinkedList<Ekintza> emaitza = ep.ekintzakHautatu(ekintzak);
        for(Ekintza e : emaitza){
            System.out.println(e.getHasieraOrdua().toString()+"
"+e.getAmaieraOrdua().toString());
        }
    }
    public static Date StringToDate(String s){
        Date result = null;
        try{
            SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
            result = dateFormat.parse(s);
        }catch(ParseException e){
            e.printStackTrace();
        }
        return result ;
    }
}

```

```

public class Ekintza {
    private Date hasieraOrdua;
    private Date amaieraOrdua;

    public Ekintza(Date hasieraOrdua, Date amaieraOrdua) {
        this.hasieraOrdua = hasieraOrdua;
        this.amaieraOrdua = amaieraOrdua;
    }
    public Date getHasieraOrdua() {
        return hasieraOrdua;
    }
    public void setHasieraOrdua(Date hasieraOrdua) {
        this.hasieraOrdua = hasieraOrdua;
    }
    public Date getAmaieraOrdua() {
        return amaieraOrdua;
    }
    public void setAmaieraOrdua(Date amaieraOrdua) {
        this.amaieraOrdua = amaieraOrdua;
    }
}

```

Analisia

N = Ekintza kopurua

- Ordenatzea MergeSort bidez egiten da. Beraz: $O(N \lg N)$
- Ekintzen prozesaketa (begiztan) lineala da ekintza kopuruan (hau da, hautagai kopuruan)

Beraz, guztira: $N \lg N + N = O(N \lg N)$