```scala
package com.test.spark3

import org.apache.spark.storage.StorageLevel
import org.apache.spark.{HashPartitioner, SparkConf, SparkContext}

object visitCountProv {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("test").setMaster("local")
    conf.set("spark.testing.memory", "471859200")
    val sc = new SparkContext(conf)   //Driver类

    val rdd = sc.textFile("D:\\doc\\study\\51CTO\\大数据中级\\day6-spark3\\电商流量数据文件\\data\\2015082818")
      .cache()        //缓存在内存和HDFS的/tmp/Spark*  路径下
      .filter(line => line.length>0)
      .map { line=>
        val arr = line.split("\\t")
        val url = arr(1)
        val guid = arr(5)
        val provinceId = arr(23)
        val date = arr(17).substring(0,10)    //2015-08-28 18:10:00
        ((date,provinceId, guid),url)
      }.filter( line=> line._2.length>5)   // length(url)>5
      .partitionBy(new HashPartitioner(10))
      .persist(StorageLevel.DISK_ONLY)

      /*
        select date, provinceId,count(url) pv ,count(distinct guid) uv
        from track_log
        group by date,provinceId
        */

    rdd.mapValues(url=>1)   // ( (date,provinceId, guid), 1)
      .reduceByKey(_+_)       // ((date,provinceId, guid), pv) pv=count(url)
      .map{  case ((date, provId, guid),pv) => ((date,provId),(pv,1))  }   //  ((date,provinceId), (pv, uv))
      .reduceByKey((x,y)=>(x._1+y._1, x._2+y._2))
      .sortByKey(true)
      .foreach(println)

    rdd.unpersist()
    sc.stop()
  }
}

```