

- `dir xyz` means that the current directory contains a directory named `xyz`.

Given the commands and output in the example above, you can determine that the filesystem looks visually like this:

```
- / (dir)
- a (dir)
  - e (dir)
    - i (file, size=584)
  - f (file, size=29116)
  - g (file, size=2557)
  - h.lst (file, size=62596)
- b.txt (file, size=14848514)
- c.dat (file, size=8504156)
- d (dir)
  - j (file, size=4060174)
  - d.log (file, size=8033020)
  - d.ext (file, size=5626152)
  - k (file, size=7214296)
```

Here, there are four directories: `/` (the outermost directory), `a` and `d` (which are in `/`), and `e` (which is in `a`). These directories also contain files of various sizes.

Since the disk is full, your first step should probably be to find directories that are good candidates for deletion. To do this, you need to determine the **total** size of each directory. The total size of a directory is the sum of the sizes of the files it contains, directly or indirectly. (Directories themselves do not count as having any intrinsic size.)

The total sizes of the directories above can be found as follows:

- The total size of directory `e` is 584 because it contains a single file `i` of size 584 and no other directories.
- The directory `a` has total size 94853 because it contains files `f` (size 29116), `g` (size 2557), and `h.lst` (size 62596), plus file `i` indirectly (`a` contains `e` which contains `i`).
- Directory `d` has total size 24933642.
- As the outermost directory, `/` contains every file. Its total size is 48381165, the sum of the size of every file.

To begin, find all of the directories with a total size of at most 100000, then calculate the sum of their total sizes. In the example above, these directories are `a` and `e`; the sum of their total sizes is 95437 (94853 + 584). (As in this example, this process can count files more than once!)

Find all of the directories with a total size of at most 100000. What is the sum of the total sizes of those directories?

Your puzzle answer was 1723892.

The first half of this puzzle is complete! It provides one gold star: \*

--- Part Two ---

Now, you're ready to choose a directory to delete.

The total disk space available to the filesystem is 70000000. To run the update, you need unused space of at least 30000000. You need to find a directory you can delete that will free up enough space to run the update.

In the example above, the total size of the outermost directory (and thus the total amount of used space) is 48381165; this means that the size of

the unused space must currently be 21618835, which isn't quite the 30000000 required by the update. Therefore, the update still requires a directory with total size of at least 8381165 to be deleted before it can run.

To achieve this, you have the following options:

- Delete directory e, which would increase unused space by 584.
- Delete directory a, which would increase unused space by 94853.
- Delete directory d, which would increase unused space by 24933642.
- Delete directory /, which would increase unused space by 48381165.

Directories e and a are both too small; deleting them would not free up enough space. However, directories d and / are both big enough! Between these, choose the **smallest**: d, increasing unused space by 24933642.

Find the smallest directory that, if deleted, would free up enough space on the filesystem to run the update. What is the total size of that directory?

Answer:  [Submit]

Although it hasn't changed, you can still [get your puzzle input](#).

You can also [\[Share\]](#) this puzzle.