The FoF Flowchart

Read N-body Header, allocate **rra,
**vva and fill these using read_output()

Copy rra and vva to
struct particle *data;

Make
grid=****allocate_particle_pointer_3d();

Attach particles to Grid Linked list

For all grid points(i,j,k=0 to N-1)

while
grid[i][j][k]
!= NULL

no

yes

cluster=grid[i][j][k]
and findfriends();

if count ≥
Nmin Print
Cluster
Position and
Mass

Print Total Elements, To-
tal Cluster and DM_m

Read N-body header, allocate **rra and **vva, and
fill these using read_output()

struct particle { float x[3], v[3]; struct particle *next;}

Copy rra,vva to struct particle data(MM)

Make grid=***allocate_particle_pointer_3D();
Attach particles to grid;

For all grid points (i,j,k=0;N-1)

Grid with particles on linked lists

NULL

NULL

If count >Nmin, print cluster positions

While grid[i,j,k]!=NULL

Yes

Clster=Grid[i,j,k];

findfriend();

Print no of clusters, etc.

```
┌─────────────────────────────────────────┐
│ cctest -> current particle in cluster    │
│ tail -> last particle in cluster         │
└─────────────────────────────────────────┘
                    │
                    ▼
        ┌──────────────────────────┐
        │  set   tail = cctest = cluster │
        └──────────────────────────┘
                    │
                    ▼
              ╱───────────╲        No        ┌──────────────────┐
             ╱   while      ╲ ──────────────▶ │ RETURN CONTROL   │
             ╲ cctest != NULL╱                └──────────────────┘
              ╲───────────╱
                    │ yes
                    ▼
             ┌──────────┐
             │ Count ++ │
             └──────────┘
                    │
                    ▼
    ┌───────────────────────────────────────────┐
    │ Determine grid of cctest (say grid[m][n][o]) │
    └───────────────────────────────────────────┘
                    │
                    ▼
              ╱───────────╲   yes    ┌───────────────────────────┐
             ╱     For      ╲───────▶ │ i = ((m + u) + N1) % N1    │
             ╲ U,v,w = -1,0,1╱        │ j = ((n + v) + N2) % N2    │
              ╲───────────╱           │ k = ((o + w) + N3) % N3    │
                    │ no              └───────────────────────────┘
                    ▼                              │
    ┌──────────────────────┐                       ▼
    │ cctest = cctest->next ; │        ┌─────────────────────────────────────┐
    └──────────────────────┘          │ gctest -> current particle in grid   │
                                      │ gptest -> previous particle in grid  │
                                      └─────────────────────────────────────┘
                                                   │
                                                   ▼
                                      ┌─────────────────────┐
                                      │        Set          │
                                      │ gctest = grid[i][j][k] ; │
                                      │   gptest = NULL ;    │
                                      │     FLAG = 0 ;       │
                                      └─────────────────────┘
                                                   │
                                                   ▼
                              no             ╱───────────╲
                     ◀────────────────────  ╱    while     ╲ ◀───────────
                                            ╲  gctest != null╱
                                             ╲───────────╱
                                                   │ yes
                                                   ▼
  ┌─────────────────────┐      no          ╱───────────╲    yes    ┌──────────────┐
  │   FLAG = 1034 ;     │ ◀────────────── ╱     if        ╲ ──────▶ │ move (i,j,k) │
  │   gptest = gctest ; │                 ╲ dist (gctest,cctest)╱    └──────────────┘
  │ gctest = gctest->next ;│               ╲  < l_fof    ╱                │
  └─────────────────────┘                  ╲───────────╱                 ▼
                                                            ╱───────────╲   yes   ┌────────────────────────────────────┐
                                                           ╱     if        ╲────▶ │ grid[p][q][r] = (grid[p][q][r])->next ; │
                                                           ╲  FLAG == 0    ╱      │   gctest->next = NULL ;            │
                                                            ╲───────────╱         │    tail->next = gctest ;          │
                                                                 │ no             │    tail = tail->next ;            │
                                                                 ▼                │  gctest = grid[p][q][r] ;         │
                                                    ┌──────────────────────┐      └────────────────────────────────────┘
                                                    │  tail->next = gctest ; │
                                                    │ gptest->next = gctest->next; │
                                                    │  gctest->next = NULL ; │
                                                    │    tail = tail->next ; │
                                                    │ gctest = gptest->next ; │
                                                    └──────────────────────┘
                                                                 │
                                                                 ▼
                                                    ┌──────────────────┐
                                                    │  Return control  │
                                                    └──────────────────┘
```

FLOW CHART FOR
FUNCTION "FIND FRIENDS "