```
In [2]:
            from daphne import daphne
            import os, json
            import numpy as np
            import torch
            from torch import tensor
            import pandas as pd
            import matplotlib.pyplot as plt
 In [3]:
            def ast helper(fname, directory):
                 sugared_fname = '../prob_prog/hw/hw6/CS532-HW6/{}/.format(directory, fname)
                 desugared_ast_json_fname = '/Users/gw/repos/prob_prog/' + sugared_fname.replace('.daphne','.json')
                 if os.path.isfile(desugared_ast_json_fname):
                      with open(desugared_ast_json_fname) as f:
                          ast = json.load(f)
                 else:
                      #note: the sugared path that goes into daphne desugar should be with respect to the daphne path!
                      ast = daphne(['desugar-hoppl-cps', '-i', sugared_fname])
                      with open(desugared_ast_json_fname, 'w') as f:
                          json.dump(ast, f)
                 return ast
            i=1
            fname = '{}.daphne'.format(i)
            exp = ast helper(fname, directory='programs')
            %cat programs/1.daphne
            (defn until-success [p n]
               (if (sample (flip p))
                 (until-success p (+ n 1))))
            (let [p 0.01]
              (until-success p 0))
 In [8]:
            import smc, evaluator
            import importlib
            importlib.reload(smc)
           <module 'smc' from '/Users/gw/repos/prob prog/hw/hw6/CS532-HW6/smc.py'>
 Out[8]:
In [11]:
            output = lambda x: x
            evaluator.evaluate(exp, env=None)('addr start', output)
Out[11]: (<function primitives.push_addr(alpha, value, k)>,
             ['addr_start', '0', <evaluator.Procedure at 0x135e73640>],
             {'type': 'proc'})
In [14]:
            n particles=3
            logZ, particles = smc.SMC(n particles, exp)
            particles
           In SMC step 0, Zs: []
           [tensor(42), tensor(122), tensor(88)]
Out[14]:
In [15]:
            particle counts = [1,10,100,1000,10000,100000]
            fig, axes = plt.subplots(nrows=len(particle counts),figsize=(30,20))
            # fig.tight layout()
            plt.subplots adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.5) # https://stackoveri
            for idx, n particles in enumerate(particle counts):
                 logZ, particles = smc.SMC(n_particles, exp)
                 samples_array = np.array([sample.item() for sample in particles])
                 mean = samples_array.mean()
                 var = samples array.var()
                 pd.Series(samples_array).plot.hist(ax=axes[idx], bins=50, title='Program {} | {} particles | mean {:1.3f}
           In SMC step 0, Zs:
           In SMC step 0, Zs: []
           In SMC step 0, Zs: []
                                                    Program 1.daphne | 1 particles | mean 457.000 | var / std 0.000e+00 / 0.000e+00 | Evidence: logZ 0.000 / Z 1.000e+00
             0.8
            0.6
            D 0.4
             0.0
                                                    Program 1.daphne | 10 particles | mean 63.200 | var / std 4.400e+03 / 6.633e+01 | Evidence: logZ 0.000 / Z 1.000e+00
             1.5
             声 1.0
                                                    ogram 1.daphne | 100 particles | mean 100.150 | var / std 8.675e+03 / 9.314e+01 | Evidence: logZ 0.000 / Z 1.000e+00
             12.5
                                                   Program 1.daphne | 1000 particles | mean 93.865 | var / std 9.031e+03 / 9.503e+01 | Evidence: logZ 0.000 / Z 1.000e+00
             150
            Frequen
100
                                                   Program 1.daphne | 10000 particles | mean 99.084 | var / std 1.031e+04 / 1.015e+02 | Evidence: logZ 0.000 / Z 1.000e+00
            )
1000
                                                   Program 1.daphne | 100000 particles | mean 98.877 | var / std 9.891e+03 / 9.945e+01 | Evidence: logZ 0.000 / Z 1.000e+00
```