```python
# Import modules
from particle_info import *
import numpy as np
import matplotlib.pyplot as plt

# Create particle classes
quark = quark()
gluon = gluon()
boson = boson()
pion = pion()
lepton = lepton()
X = X()
neutrino = neutrino()
photon = photon()

# Define total g in this zoo of particles
total_g = photon.g() + lepton.e.g() + lepton.muon.g() + pion.pi_pm.g() + pion.pi_0.g() + X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g() + gluon.g() + lepton.tau.g()


def g_star_s(T):
    '''

    Parameters
    ----------
    T : Temperature

    Returns
    -------
    g_star at some value of T

    '''
    g_star_max = total_g + neutrino.g_s()

    if T > X.T():
        return g_star_max

    if X.T() ⩾ T > quark.top.T():
        g_minus = X.g()
        return g_star_max - g_minus

    if quark.top.T() ⩾ T > boson.W.T():
        g_minus = X.g() + quark.top.g()
        return g_star_max - g_minus

    if boson.W.T() ⩾ T > quark.bottom.T():
        g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g()
        return g_star_max - g_minus

    if quark.bottom.T() ⩾ T > pion.pi_pm.T_join():
        g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g()
        return g_star_max - g_minus

    if pion.pi_pm.T_join() ⩾ T > pion.pi_pm.T_leave():
```

```
54          g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g
    () + quark.bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() +
    quark.down.g() + gluon.g() + lepton.tau.g()
55          g_plus = pion.pi_pm.g() + pion.pi_0.g()
56          return g_star_max - g_minus + g_plus
57
58      if pion.pi_pm.T_leave() ⩾ T > lepton.e.T():
59          g_minus = lepton.muon.g() + pion.pi_pm.g() + pion.pi_0.g() + X.g() +
    quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g() +
    quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g() + gluon.g
    () + lepton.tau.g()
60          return g_star_max - g_minus
61
62      if T ⩽ lepton.e.T():
63          g_minus = lepton.e.g() + lepton.muon.g() + pion.pi_pm.g() + pion.pi_0.g
    () + X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.
    bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g
    () + gluon.g() + lepton.tau.g()
64          return g_star_max - g_minus
65
66
67  def g_star(T):
68      '''
69
70      Parameters
71      ----------
72      T : Temperature
73
74      Returns
75      -------
76      g_star_S at some value of T
77
78      '''
79
80      g_star_max = total_g + neutrino.g()
81
82      if T > X.T():
83          return g_star_max
84
85      if X.T() ⩾ T > quark.top.T():
86          g_minus = X.g()
87          return g_star_max - g_minus
88
89      if quark.top.T() ⩾ T > boson.W.T():
90          g_minus = X.g() + quark.top.g()
91          return g_star_max - g_minus
92
93      if boson.W.T() ⩾ T > quark.bottom.T():
94          g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g()
95          return g_star_max - g_minus
96
97      if quark.bottom.T() ⩾ T > pion.pi_pm.T_join():
98          g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g
    () + quark.bottom.g()
99          return g_star_max - g_minus
100
101     if pion.pi_pm.T_join() ⩾ T > pion.pi_pm.T_leave():
```

```
        g_minus = X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g() + gluon.g() + lepton.tau.g()
        g_plus = pion.pi_pm.g() + pion.pi_0.g()
        return g_star_max - g_minus + g_plus

    if pion.pi_pm.T_leave() ⩾ T > lepton.e.T():
        g_minus = lepton.muon.g() + pion.pi_pm.g() + pion.pi_0.g() + X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g() + gluon.g() + lepton.tau.g()
        return g_star_max - g_minus

    if T ⩽ lepton.e.T():
        g_minus = lepton.e.g() + lepton.muon.g() + pion.pi_pm.g() + pion.pi_0.g() + X.g() + quark.top.g() + boson.W.g() + boson.Z.g() + boson.h.g() + quark.bottom.g() + quark.charm.g() + quark.strange.g() + quark.up.g() + quark.down.g() + gluon.g() + lepton.tau.g()
        return g_star_max - g_minus


# Define printing array and print table
T_print = np.array([200, 50, 20, 10, 0.5, 0.1, 0.0001]) * 1000
print("(T,g_star,g_star_s)")
for T in T_print:
    print("{0},{1},{2}".format(T / 1000, round(g_star(T), 2), round(g_star_s(T), 2)))

# Define thresholds
T_threshold = [100, 30, 15, 1, 0.2, 0.05, 0.00025]  # GeV

# Define T space
T = np.arange(start=0.01, stop=200 * 1000, step=0.01)  # MeV

# Solve
g_star_array = [g_star(T_val) for T_val in T]
g_star_s_array = [g_star_s(T_val) for T_val in T]

# plot
for T_val in T_threshold:
    plt.vlines(T_val, 0, 200, linestyles='dashed', colors='k', linewidth=0.75)
plt.loglog(T / 1000, g_star_array, 'k-', linewidth=1)
plt.loglog(T / 1000, g_star_s_array, 'r-', linewidth=1)
plt.xlabel(r'$T$ (GeV)')
plt.ylabel(r'$g_{\star}(T)$')
plt.axis([max(T / 1000), min(T / 1000), 1, 200])
plt.legend(["$g_{\star}$", "$g_{\star S}$"])
plt.savefig(fname='g_star_T', dpi=300)
plt.show()

```